



# Expressiveness of component-based frameworks: A study of the expressiveness of BIP

Eduard Baranov, Simon Bliudze

## ► To cite this version:

Eduard Baranov, Simon Bliudze. Expressiveness of component-based frameworks: A study of the expressiveness of BIP. *Acta Informatica*, 2020, 57, pp.761–800. 10.1007/s00236-019-00337-7 . hal-02149934v2

**HAL Id: hal-02149934**

**<https://hal.science/hal-02149934v2>**

Submitted on 28 Mar 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Expressiveness of component-based frameworks

## A study of the expressiveness of BIP

Eduard Baranov · Simon Bliudze

Received: date / Accepted: date

**Abstract** We extend our previous algebraic formalisation of the notion of component-based framework in order to formally define two forms—strong and weak—of the notion of full expressiveness and study their properties. Our earlier result shows that the BIP (Behaviour-Interaction-Priority) framework does not possess the strong full expressiveness with respect to the sub-class of GSOS rules used for the definition of its semantics. In this paper we refine this comparison detailing the expressiveness of classical BIP, Offer BIP and a number of variations obtained either by relaxing the constraints in the definition of priority models, or by introducing positive premises into the rule formats used to define the operational semantics of composition operators. The obtained results are organised into an expressiveness hierarchy.

**Keywords** component-based frameworks, expressiveness, BIP, glue operators, SOS formats

### 1 Introduction

In our previous work [4], we have formalised some of the properties that are desirable for component-based design frameworks, namely: incrementality, flattening, compositionality and modularity [25, 35]. The formalisation is based on a very simple, abstract algebraic definition of the notion of component-based framework. We have also discussed the full expressiveness property, although without providing a formal definition for it. Intuitively, *strong (resp. weak) full expressiveness of one framework w.r.t. another* requires that each operator of the first be expressible as an operator (resp. composition of operators) in the second.

---

Eduard Baranov  
Université catholique de Louvain, Place de l'Université 1,  
1348 Ottignies-Louvain-la-Neuve, Belgium  
E-mail: eduard.baranov@uclouvain.be      ORCID: 0000-0002-7357-705X

Simon Bliudze  
INRIA Lille – Nord Europe, 40 avenue Halley, 59650 Villeneuve d'Ascq, France  
Tel.: +33 3 59 35 86 31  
E-mail: simon.bliudze@inria.fr      ORCID: 0000-0002-7900-5271

In this paper, we provide a formal definition of the notion of full expressiveness. We then apply it to study the expressiveness of the BIP (Behaviour-Interaction-Priority) framework [7,8] and their variations.

BIP is a component-based framework for the design of concurrent systems based on the separation of concerns between coordination and computation [7,8]. It is among the few frameworks, such as SCADE [1] and Ptolemy II [20], which combine the use of formal methods for defining their operational semantics and for model verification, with code generation. Recently we have developed a *theory of architectures* [2,3], which allows the compositional design of correct-by-construction BIP models. This theory formalises BIP design patterns, called *architectures*. The application of an architecture to a BIP system enforces the corresponding *characteristic property*. We have shown that safety properties are preserved by composition of architectures and have provided an algorithm for verifying whether two architectures are non-interfering, which guarantees the preservation of their liveness properties. This approach provides a compositional method for the design of correct-by-construction systems from a set of formalised user and system requirements. We have applied this methodology for the design of satellite on-board software, in particular for CubETH—a university-built nanosatellite—and in a project financed by the European Space Agency [28,37].

BIP systems consist of components modelled as Labelled Transition Systems (LTS). Transitions are labelled by ports, which are used for synchronisation with other components. Composition operators defining such synchronisations are obtained by combining *interaction* and *priority models*. The classical operational semantics of the BIP composition operators [10] is defined by SOS rules in a format, which is a restriction of GSOS [15]; we call this format *BIP-like SOS*. In the previous works [10,11,12,13], we have conducted an extensive study of the semantics and algebraic representations of the BIP glue operators. While the semantics of interaction models is very straightforward and did not change throughout those papers, that of the priority models has proven to be much subtler. Thus, in order to make the correct decisions for the future versions and flavours of BIP, it is essential to understand the impact of the meaning given to priority models on the resulting design framework, in particular its expressiveness, which is the central question of this paper.

In [4], we have provided a counter-example showing that the classical semantics of BIP does not possess flattening, which implies that it does not possess strong full expressiveness w.r.t. BIP-like SOS either. This shows that the often encountered informal statement: “BIP possesses the expressiveness of the universal glue” (or its equivalent in slightly different formulations) is based on an erroneous proposition in previous work [11, Proposition 4]. The fundamental reasons for this absence of strong full expressiveness lie in the definition of the priority models. A priority model is a strict partial order on the underlying interaction model (set of allowed interactions). In particular, this definition guarantees that applying a priority model does not introduce deadlocks in the otherwise deadlock-free system. This property turns out to be one of the key reasons underlying the expressiveness limitations, since deadlocks can be introduced by certain operators respecting the BIP-like SOS format.

In this paper, we consider five variations of BIP, including the classical BIP, alongside five variations of the SOS formats defining the composition operator semantics and study their relative expressiveness. In doing so, we extend and refine

the results previously published in the proceedings of the EXPRESS/SOS 2016 workshop [5].

The first part of the paper introduces the general theory of expressiveness, by

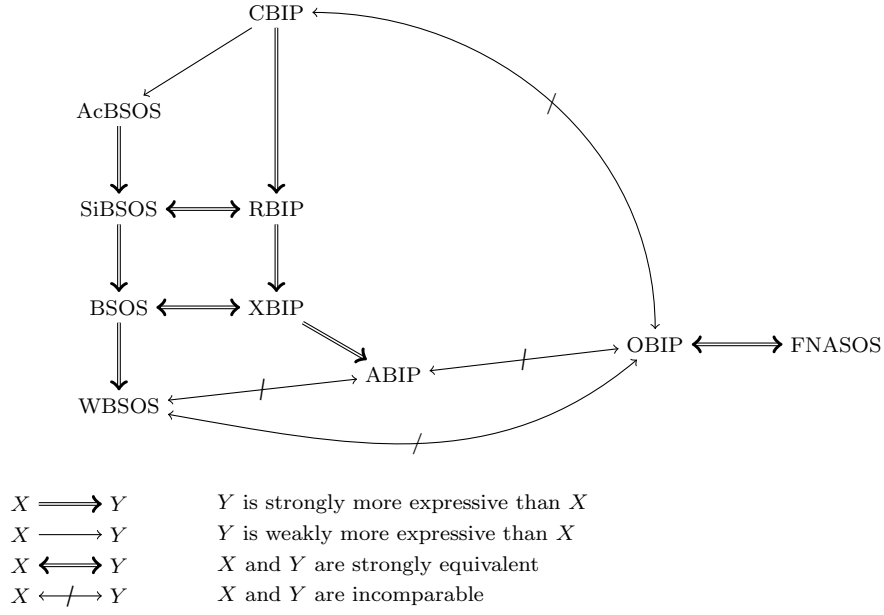
- defining the notion of component-based framework,
- identifying six possible expressiveness relations between two frameworks based on the notions of strong and weak full expressiveness of one framework w.r.t. another,
- presenting several properties of these relations.

The second part of the paper studies the expressiveness relations among five variations of BIP and five SOS formats:

- **Classical BIP (CBIP)** and **BIP-like SOS (BSOS)**
- **Witness BIP-like SOS (WBSOS)**: WBSOS extends BIP-like SOS with positive premises that do not contribute to the transition defined by the rule, i.e. the enabledness of a transition in one of the components is tested without the transition being fired.
- **Simple (SiBSOS)** and **acyclic (AcBSOS) BIP-like SOS**: These two restrictions of BSOS apply to sets of rules defining an operator. For a set of rules in the BIP-like SOS format, the *inhibiting relation* identifies interactions, whereof the enabledness inhibits a given interaction. SiBSOS considers only sets of BIP-like SOS rules with *simple* inhibiting relations, where the enabledness of one interaction is always sufficient to inhibit another one. AcBSOS requires in addition that the inhibiting relation be acyclic.
- **Relaxed BIP (RBIP)**: RBIP extends classical BIP by taking priority models to be arbitrary relations on sets of ports, i.e. without requiring that these be strong partial orders and limited to interactions provided by the components.
- **Complex BIP (XBIP)**: XBIP is a further extension of RBIP, which allows sets of interactions—as opposed to single interactions in CBIP and RBIP—to be used as inhibitors in the priority model.
- **Offer BIP (OBIP)** and **Firing-Negative-Activation SOS (FNASOS)**: These two frameworks are based on an alternative, offer semantics initially introduced in [13], using the offer predicate, which allows testing whether a port is part of an enabled transition in an atomic component.
- **Activation BIP (ABIP)**: ABIP is a hybrid framework introduced for the sake of comparison, which mixes the classical and offer semantics by relying on the usual transition relation to define the semantics of priority models, but using the offer predicate for that of non-firing positive premises.

Figure 1 shows the comparison of expressiveness among these frameworks. The figure is complete in the sense that all relations that are not shown explicitly can be deduced from those shown by application of the results presented in Section 2.

The rest of the paper is structured as follows. Section 2 presents the algebraic formalisation of component-based frameworks, strong and weak full expressiveness, and presents their properties. Section 3 provides the definitions and the formal semantics of all the variations of BIP and the SOS formats presented above. Section 4 analyses the expressiveness relations among these frameworks. Section 5 briefly discusses some related work. Finally, Section 6 concludes the paper.



**Fig. 1** Expressiveness relations among the considered frameworks.

## 2 Algebraic formalisation of component-based frameworks

### 2.1 Basic definitions

Every component-based design framework can be viewed as an algebra of components equipped with a semantic mapping. The algebra of components *syntactically* defines the composite components that can be assembled from a given set of the atomic ones. The semantic mapping associates to each component its corresponding behaviour. The codomain of the semantic mapping, which we call the *semantic domain* consists of a behaviour type—defined in terms of Labelled Transition Systems or a similar formalism—and an associated equivalence relation. This can be formalised as follows:

**Definition 1** A *component-based framework* is a tuple  $(\mathbf{G}, \mathbf{C}, \mathbf{B}, \approx, \sigma)$ , where

- $\mathbf{G}$  is a set of *composition (glue) operators*, we denote by  $\mathbf{G}^n \subseteq \mathbf{G}$ , with  $n \in \mathbb{N}$ , the subset of  $n$ -ary operators,
- $\mathbf{C}$  is a *set of atomic components*,
- $(\mathbf{B}, \approx)$  is a *semantic domain*, consisting of a *behaviour type*  $\mathbf{B}$  and an *equivalence relation*  $\approx \subseteq \mathbf{B} \times \mathbf{B}$ ,
- $\sigma : \mathbf{A} \rightarrow \mathbf{B}$  is a *partial semantic mapping* from the algebraic structure

$$\mathbf{A} ::= C \mid o\langle C_1, \dots, C_n \rangle, \quad C \in \mathbf{C}, \quad n \in \mathbb{N}, \quad C_1, \dots, C_n \in \mathbf{A} \text{ and } o \in \mathbf{G}^n,$$

generated by  $\mathbf{G}$  from  $\mathbf{C}$ , which we call the *algebra of components* of the framework. (Notice that  $\mathbf{A}$  does not appear explicitly in the tuple, since it is fully defined by  $\mathbf{G}$  and  $\mathbf{C}$ .)

We call the elements of  $\mathbf{A}$  *components* and the elements of  $\mathbf{B}$  *behaviours*. The algebraic structure  $\mathbf{A}$  represents the set of all systems constructible within the framework.

The behaviour type  $\mathbf{B}$  defines the semantic nature of the components manipulated by the framework. The equivalence relation  $\simeq \subseteq \mathbf{B} \times \mathbf{B}$  allows comparing components in terms, for example, of their functionality, observable behaviour or capability of interaction with the environment. It is canonically lifted to  $\mathbf{A}$  by putting,  $C_1 \simeq C_2$  iff  $\sigma(C_1) \simeq \sigma(C_2)$ , for any  $C_1, C_2 \in \mathbf{A}$ . The semantic mapping  $\sigma : \mathbf{A} \rightarrow \mathbf{B}$  assigns to each component its meaning in terms of the behaviour type  $\mathbf{B}$ : for any  $C \in \mathbf{A}$ , we say that  $\sigma(C)$  is the behaviour of  $C$ .

*Example 1* A number of examples representing variations of the BIP framework will be provided in the subsequent sections. For an alternative example, consider the framework  $\text{CCS}^- = (\mathbf{G}^-, \mathbf{C}, \mathbf{B}, \simeq, \sigma^-)$ , taking both  $\mathbf{C}$  and  $\mathbf{B}$  to be the subset of purely sequential processes in CCS [29]:

$$\mathbf{C} = \mathbf{B} ::= l \mid l.P \mid P_1 + P_2, \quad l \in \mathcal{L}, \quad P, P_1, P_2 \in \mathbf{B},$$

where  $\mathcal{L} = \{\tau\} \cup \{a, \bar{a} \mid a \in \mathcal{A}\}$ , for some given set of actions  $\mathcal{A}$ . Take  $\simeq$  be the branching bisimilarity relation [6]. Take  $\mathbf{G} = \{\parallel, \backslash A\}$ , where  $\parallel$  is the classical binary CCS parallel composition (synchronising  $a$  and  $\bar{a}$ , for any  $a \in \mathcal{A}$ ) and  $\backslash A$  is the unary restriction operator, which hides all actions in the set  $A \subseteq \mathcal{A}$  by replacing them with  $\tau$ . The semantic mapping  $\sigma$  is defined trivially for  $\backslash A$  and through the expansion lemma [29], for  $\parallel$ .

**Definition 2** The semantic mapping is called *structural*, if it is defined by associating to each  $n$ -ary glue operator  $o \in \mathbf{G}^n$  a corresponding partial mapping  $\tilde{o} : \mathbf{B}^n \rightarrow \mathbf{B}$  and putting

$$\sigma(o\langle C_1, \dots, C_n \rangle) = \tilde{o}(\sigma(C_1), \dots, \sigma(C_n)), \text{ for all } n \in \mathbb{N}, C_1, \dots, C_n \in \mathbf{A} \text{ and } o \in \mathbf{G}^n.$$

We call  $\{\tilde{o} \mid o \in \mathbf{G}\}$  the set of *defining mappings* of  $\sigma$ .

*Example 2* Clearly, the semantic mapping in Example 1 is structural. Alternatively, any semantic mapping defined using Structural Operational Semantics rules [33] is, indeed, structural.

Before moving on to the notions necessary for comparing the expressiveness of component-based frameworks, we introduce the following technical definition.

**Definition 3** Given a framework  $F = (\mathbf{G}, \mathbf{C}, \mathbf{B}, \simeq, \sigma)$  and a set of variables  $\mathcal{Z}$ , we will denote by  $\mathbf{G}[\mathcal{Z}]$  the set of *expressions on variables in  $\mathcal{Z}$* , defined by the following grammar:

$$\mathbf{G}[\mathcal{Z}] ::= Z \mid o\langle E_1, \dots, E_n \rangle, \quad Z \in \mathcal{Z}, \quad n \in \mathbb{N}, \quad E_1, \dots, E_n \in \mathbf{G}[\mathcal{Z}] \text{ and } o \in \mathbf{G}^n.$$

Comparing the expressiveness of two component-based frameworks is only possible when their semantic domains coincide.<sup>1</sup>

<sup>1</sup> Two component-based frameworks with distinct semantic domains can be compared by mapping to a common behaviour type and taking an appropriate equivalence relation consistent with those of the frameworks. However, this essentially boils down to a substitution of the semantic domains, i.e. considering a different pair of frameworks.

**Definition 4** Given two frameworks  $F_i = (\mathbf{G}_i, \mathbf{C}_i, \mathbf{B}, \approx, \sigma_i)_{i \in \{1,2\}}$  with the same semantic domain, we say that  $F_1$  has *strong full expressiveness w.r.t.  $F_2$* , denoted  $F_2 \blacktriangleleft F_1$  iff

$$\forall n \in \mathbb{N}, \forall o \in \mathbf{G}_2^n, \exists \tilde{o} \in \mathbf{G}_1^n : \forall C_1^i, \dots, C_n^i \in \mathbf{A}_i, \\ \bigwedge_{k=1}^n \sigma_1(C_k^1) \approx \sigma_2(C_k^2) \implies \sigma_1(\tilde{o}(C_1^1, \dots, C_n^1)) \approx \sigma_2(o(C_1^2, \dots, C_n^2)). \quad (1)$$

We say that  $F_1$  has *weak full expressiveness w.r.t.  $F_2$* , denoted  $F_2 \triangleleft F_1$  iff,

$$\forall n \in \mathbb{N}, \forall o \in \mathbf{G}_2^n, \exists \tilde{o} \in \mathbf{G}_1[Z_1, \dots, Z_n] : \forall C_1^i, \dots, C_n^i \in \mathbf{A}_i, \\ \bigwedge_{k=1}^n \sigma_1(C_k^1) \approx \sigma_2(C_k^2) \implies \sigma_1(\tilde{o}[C_1^1/Z_1, \dots, C_n^1/Z_n]) \approx \sigma_2(o(C_1^2, \dots, C_n^2)), \quad (2)$$

where  $\tilde{o}[C_1^1/Z_1, \dots, C_n^1/Z_n] \in \mathbf{A}_1$  is the component obtained by substituting in  $\tilde{o}$  the variables  $Z_k$  by components  $C_k^1$ , for all  $k \in [1, n]$ .

*Example 3* In addition to  $CCS^-$  from Example 1, consider the framework  $CCS = (\mathbf{G}, \mathbf{C}, \mathbf{B}, \approx, \sigma)$ , where  $\mathbf{C}, \mathbf{B}$  and  $\approx$  are the same as in  $CCS^-$ , whereas  $\mathbf{G} = \mathbf{G}^- \cup \{+\}$ , with the extension from  $\sigma^-$  to  $\sigma$  being trivial:  $\sigma(C_1 + C_2) \stackrel{\text{def}}{=} \sigma(C_1) + \sigma(C_2)$ . We trivially have  $CCS^- \blacktriangleleft CCS$ .

It is easy to show that the  $+$  operator cannot be encoded by any combination of parallel composition and restriction, essentially since the choice represented by  $+$  has to be maintained throughout the subsequent execution of the process, whereas both parallel composition and restriction are “memoryless”. Hence,  $CCS \not\blacktriangleleft CCS^-$ .

Notice that both strong and weak full expressiveness are preorders, i.e. they are reflexive and transitive. Strong full expressiveness trivially implies the weak one.

**Definition 5** Based on the full expressiveness relations, we introduce six comparison relations, presented in Table 1. For instance (second cell of the first row), if  $F_1$  has strong full expressiveness w.r.t.  $F_2$ , whereas  $F_2$  has only weak—but not strong—full expressiveness w.r.t.  $F_1$ , we say that  $F_1$  is *weakly more expressive than  $F_2$*  (alternatively,  $F_2$  is *weakly less expressive than  $F_1$* ) and denote this by  $F_2 \rightarrow F_1$ .

*Example 4* The full expressiveness relations from Example 3 mean that  $CCS$  is strongly more expressive than  $CCS^-$ , i.e.  $CCS^- \Rightarrow CCS$ .

Notice that the six comparison relations are mutually disjoint.

The intuitive meanings of the three relations in the first row of Table 1 are the following: given two frameworks  $F_i = (\mathbf{G}_i, \mathbf{C}_i, \mathbf{B}, \approx, \sigma_i)$ , if for any operator in  $\mathbf{G}_2$  we can find a corresponding operator in  $\mathbf{G}_1$  such that its application to an equivalent set of components would result in equivalent components, then  $F_1$  has at least equivalent expressiveness or is more expressive than  $F_2$ . There are three options for the converse. If there is a corresponding operator in  $\mathbf{G}_2$  for any operator in  $\mathbf{G}_1$ , then their expressiveness are equivalent. If every operator in  $\mathbf{G}_1$ , which does not have a corresponding one in  $\mathbf{G}_2$ , can be represented by a hierarchy of operators in

**Table 1** Expressiveness comparison relations (three “less expressive” relations indicated in grey are symmetric to the “more expressive” ones; the two relations in the second row are not used in the subsequent sections of this paper—they are only presented for the sake of completeness)

$F_1 ? F_2$ $F_2 ? F_1$	$\blacktriangleleft$	$\blacktriangleleft \wedge \blacktriangleright$	$\blacktriangleright$
$\blacktriangleleft$	$F_1$ and $F_2$ have strongly equivalent expressiveness $F_1 \Leftrightarrow F_2$	$F_1$ is weakly more expressive than $F_2$ $F_2 \rightarrow F_1$	$F_1$ is strongly more expressive than $F_2$ $F_2 \Rightarrow F_1$
$\blacktriangleleft \wedge \blacktriangleright$	$F_1$ is weakly <i>less</i> expressive than $F_2$ $F_2 \leftarrow F_1$	$F_1$ and $F_2$ have weakly equivalent expressiveness $F_1 \leftrightarrow F_2$	$F_1$ is slightly more expressive than $F_2$ $F_2 \dashrightarrow F_1$
$\blacktriangleright$	$F_1$ is strongly <i>less</i> expressive than $F_2$ $F_2 \Leftarrow F_1$	$F_1$ is slightly <i>less</i> expressive than $F_2$ $F_2 \Leftarrow F_1$	$F_1$ and $F_2$ are incomparable $F_1 \nleftrightarrow F_2$

$\mathbf{G}_2$ , then  $F_1$  is weakly more expressive than  $F_2$ . Finally, if there exists an operator in  $\mathbf{G}_1$  that cannot be represented by any combination of operators in  $\mathbf{G}_2$ , then  $F_1$  is strongly more expressive than  $F_2$ . If such inexpressible operators exist in both  $\mathbf{G}_1$  and  $\mathbf{G}_2$  then  $F_1$  and  $F_2$  are incomparable (third row of Table 1). The intuition behind the relations in the second row of Table 1 is similar.

Notice that the relations shown in Table 1 are mutually exclusive. For instance, contrary to the usual intuition behind the use of the symbols ‘ $\Leftrightarrow$ ’ and ‘ $\Rightarrow$ ’ in predicate logics,  $F_1 \Leftrightarrow F_2$  implies  $F_1 \not\Rightarrow F_2$ . In particular, a framework is never more expressive than itself, i.e.  $F \not\Rightarrow F$  and  $F \not\rightarrow F$ .

## 2.2 Properties of the comparison relations

We now provide some key properties of the relations defined in the previous section.

The relations  $\Rightarrow$ ,  $\rightarrow$  and  $\Leftrightarrow$  are transitive.

**Proposition 1** *For any frameworks  $F_1, F_2, F_3$  and any  $\mathcal{R} \in \{\Rightarrow, \rightarrow, \Leftrightarrow\}$ , we have  $F_1 \mathcal{R} F_2$  and  $F_2 \mathcal{R} F_3$  implies  $F_1 \mathcal{R} F_3$ .*

*Proof.* We provide the proof for  $\mathcal{R} = \rightarrow$ . Those for  $\Leftrightarrow$  and  $\Rightarrow$  are even more straightforward.

Since  $F_1 \rightarrow F_2$  and  $F_2 \rightarrow F_3$ , by definition of  $\rightarrow$ , hold the relations shown in the following diagram:

$$\begin{array}{ccccc}
 F_1 & \blacktriangleleft & F_2 & \blacktriangleleft & F_3 \\
 \parallel & & \parallel & & \parallel \\
 F_1 & \blacktriangleright & F_2 & \blacktriangleright & F_3 \\
 \parallel & & \parallel & & \parallel \\
 F_1 & \blacktriangledown & F_2 & \blacktriangledown & F_3
 \end{array}$$

From the transitivity of  $\blacktriangleleft$  and  $\blacktriangleright$ , we conclude that  $F_1 \blacktriangleleft F_3$  and  $F_3 \blacktriangleright F_1$ .

Suppose that  $F_3 \blacktriangleleft F_1$ . Then, from the above diagram, we deduce  $F_2 \blacktriangleleft F_3 \blacktriangleleft F_1$ , i.e.  $F_2 \blacktriangleleft F_1$ , which contradicts the assumption  $F_1 \rightarrow F_2$ . We conclude that  $F_3 \not\blacktriangleleft F_1$  and, therefore,  $F_1 \rightarrow F_3$ .  $\square$



In particular, transitivity of  $\Leftrightarrow$  implies that this relation is, indeed, an equivalence. We will now show that all the relations introduced above are preserved by  $\Leftrightarrow$ . First, consider the following lemma.

**Lemma 1** *For any frameworks  $F_1, F_2, F_3$ , such that  $F_1 \Leftrightarrow F_2$  and any  $\mathcal{R} \in \{\blacktriangleleft, \blacktriangleright, \triangleleft, \triangleright\}$ , we have  $F_1 \mathcal{R} F_3$  iff  $F_2 \mathcal{R} F_3$ .*

*Proof.* Follows from the transitivity of  $\blacktriangleleft$  and  $\triangleleft$  and the fact that  $\blacktriangleleft$  implies  $\triangleleft$ .  $\square$

**Corollary 1** *For any frameworks  $F_1, F_2, F_3$ , such that  $F_1 \Leftrightarrow F_2$  and any  $\mathcal{R} \in \{\Leftrightarrow, \rightarrow, \Rightarrow, \Leftarrow, \rightarrow, \leftrightarrow\}$ , we have  $F_1 \mathcal{R} F_3$  iff  $F_2 \mathcal{R} F_3$ .*

**Lemma 2** *For any frameworks  $F_1, F_2, F_3$ , hold the following implications:*

1.  $F_1 \blacktriangleleft F_2 \wedge F_1 \nrightarrow F_3$  implies  $F_2 \nrightarrow F_3$ ,
2.  $F_1 \blacktriangleleft F_2 \wedge F_1 \nrightarrow F_3$  implies  $F_2 \nrightarrow F_3$ ,
3.  $F_1 \blacktriangleleft F_2 \wedge F_3 \nrightarrow F_2$  implies  $F_3 \nrightarrow F_1$ ,
4.  $F_1 \blacktriangleleft F_2 \wedge F_3 \nrightarrow F_2$  implies  $F_3 \nrightarrow F_1$ .

*Proof.* Similarly to Lemma 1, all these implications follow from the transitivity of  $\blacktriangleleft$  and  $\triangleleft$  and, for implications 2 and 4, from the fact that  $\blacktriangleleft$  implies  $\triangleleft$ .  $\square$

From Lemma 2 and the transitivity of  $\blacktriangleleft$  and  $\triangleleft$  we deduce the following corollary.

**Corollary 2** *For any frameworks  $F_1, F_2, F_3$ , hold the following implications:*

1.  $F_1 \Rightarrow F_2 \rightarrow F_3$  implies  $F_1 \Rightarrow F_3$ ,
2.  $F_1 \rightarrow F_2 \Rightarrow F_3$  implies  $F_1 \Rightarrow F_3$ .

*Proof.* In both items, we have  $F_1 \blacktriangleleft F_2 \blacktriangleleft F_3$ . From the transitivity of  $\blacktriangleleft$  we deduce  $F_1 \blacktriangleleft F_3$ .

In 1, we have  $F_2 \nrightarrow F_1 \wedge F_2 \blacktriangleleft F_3$  which, by Lemma 2 (2), implies  $F_3 \nrightarrow F_1$ .

Similarly, in 2, we have  $F_1 \blacktriangleleft F_2 \wedge F_3 \nrightarrow F_2$  which, by Lemma 2 (4), implies  $F_3 \nrightarrow F_1$ .

Thus, in both cases, we, indeed, have  $F_1 \Rightarrow F_3$ .  $\square$

Notice that relation combinations other than those in Proposition 1 and the two corollaries above do not provide immediate “shortcut” relations. This is mostly due to the fact that the complement relations  $\nrightarrow$  and  $\nrightarrow$  are not transitive. For instance, given  $F_1, F_2$  and  $F_3$ , such that  $F_1 \triangleleft F_2 \triangleleft F_3$  but  $F_1 \nrightarrow F_2 \nrightarrow F_3$ , we can proceed as follows:

1. Take all composition operators from  $F_1$  that do not have corresponding ones in  $F_2$  and add them to  $F_3$  as “syntactic sugar” for the corresponding hierarchical expressions.
2. Add two *completely new* operators two  $F_3$  and another completely new operator to  $F_2$  with the semantics defined as the composition of those for the two new operators in  $F_3$ .

Denoting the extended frameworks by  $F_2'$  and  $F_3'$ , we have  $F_1 \triangleleft F_2' \triangleleft F_3'$ ,  $F_1 \nrightarrow F_2' \nrightarrow F_3'$  but  $F_1 \blacktriangleleft F_3'$ . Without constituting a formal proof, this manipulation does provide an intuition for the reason why other relation combinations do not have generic shortcuts.

**Definition 6** The framework is said to possess *uniform flattening*<sup>2</sup> if the set of composition operators  $\mathbf{G}$  is closed under composition:

$$\begin{aligned} \forall n \in \mathbb{N}, \forall i, j \in [1, n] (i \leq j), \\ \forall o_1 \in \mathbf{G}^{n-j+i}, \forall o_2 \in \mathbf{G}^{j-i+1}, \exists o_3 \in \mathbf{G}^n, \forall C_1, C_2, \dots, C_n \in \mathbf{A} : \\ o_1 \langle C_1, \dots, C_{i-1}, o_2 \langle C_i, \dots, C_j \rangle, C_{j+1}, \dots, C_n \rangle \simeq o_3 \langle C_1, \dots, C_n \rangle. \end{aligned}$$

As mentioned above, strong full expressiveness trivially implies weak full expressiveness. The converse holds in presence of uniform flattening.

**Proposition 2** For two frameworks  $F_1$  and  $F_2$ , such that  $F_1$  has uniform flattening,  $F_2 \triangleleft F_1$  implies  $F_2 \blacktriangleleft F_1$ .

*Sketch of the proof.* Weak full expressiveness guaranties that any operator  $o$  from  $\mathbf{G}_2$  is expressible as a hierarchy of operators in  $\mathbf{G}_1$ . Uniform flattening applied several times to this hierarchy can “flatten” it to a single glue corresponding to  $o$ . Thus, the requirement for strong full expressiveness is satisfied.  $\square$

According to Definition 4, in order to establish that one framework has strong (resp. weak) full expressiveness w.r.t. another, we have to prove the existence of the corresponding operator (resp. hierarchical construction) that preserves the semantic equivalence (see (1) or (2)). Below, we show that, under additional assumptions, it is sufficient to only check the preservation of the behaviour *equality*.

**Definition 7** A framework  $(\mathbf{G}, \mathbf{C}, \mathbf{B}, \simeq, \sigma)$  is *compositional* iff

$$\begin{aligned} \forall n \in \mathbb{N}, \forall o \in \mathbf{G}^n, \forall C_1^i, \dots, C_n^i \in \mathbf{A}_i (i \in \{1, 2\}), \\ \bigwedge_{k=1}^n C_k^1 \simeq C_k^2 \implies o \langle C_1^1, \dots, C_n^1 \rangle \simeq o \langle C_1^2, \dots, C_n^2 \rangle. \end{aligned} \quad (3)$$

**Lemma 3** ([4]) If the semantic mapping of a framework is structural and the defining mappings (see Definition 2) preserve the equivalence of the operands, then the framework is compositional.

*Proof.* By the assumptions of the lemma, we have:

$$\begin{aligned} \sigma(o \langle C_1^1, \dots, C_n^1 \rangle) &= \tilde{o}(\sigma(C_1^1), \dots, \sigma(C_n^1)) \\ &\simeq \tilde{o}(\sigma(C_1^2), \dots, \sigma(C_n^2)) = \sigma(o \langle C_1^2, \dots, C_n^2 \rangle). \end{aligned}$$

$\square$

All composition operators that we will consider in this paper satisfy the assumptions of the above lemma.

**Proposition 3** For two frameworks  $F_i = (\mathbf{G}_i, \mathbf{C}, \mathbf{B}, \simeq, \sigma_i)_{i \in \{1, 2\}}$ , whereof  $F_1$  is compositional, with the same atomic components, the same semantic domain, and such that, for any  $C \in \mathbf{C}$ , holds  $\sigma_1(C) = \sigma_2(C)$ , we have

<sup>2</sup> The notion of uniform flattening is stronger than that of flattening introduced in [4] in that it requires the operator  $o_3$  to be the same, independently of the choice of  $C_1, \dots, C_n$ .

1.  $F_2 \blacktriangleleft F_1$  if

$$\forall n \in \mathbb{N}, \forall o \in \mathbf{G}_2^n, \exists \tilde{o} \in \mathbf{G}_1^n : \forall C_1^i, \dots, C_n^i \in \mathbf{A}_i, \\ \bigwedge_{k=1}^n \sigma_1(C_k^1) = \sigma_2(C_k^2) \implies \sigma_1(\tilde{o}\langle C_1^1, \dots, C_n^1 \rangle) = \sigma_2(o\langle C_1^2, \dots, C_n^2 \rangle), \quad (4)$$

2.  $F_2 \blacktriangleleft F_1$  if

$$\forall n \in \mathbb{N}, \forall o \in \mathbf{G}_2^n, \exists \tilde{o} \in \mathbf{G}_1[Z_1, \dots, Z_n] : \forall C_1^i, \dots, C_n^i \in \mathbf{A}_i, \\ \bigwedge_{k=1}^n \sigma_1(C_k^1) = \sigma_2(C_k^2) \implies \sigma_1(\tilde{o}[C_1^1/Z_1, \dots, C_n^1/Z_n]) = \sigma_2(o\langle C_1^2, \dots, C_n^2 \rangle), \quad (5)$$

where all notations are as in Definition 4.

*Proof.* We prove the proposition for  $F_2 \blacktriangleleft F_1$ —the proof for  $F_2 \triangleleft F_1$  is similar.

Let us denote  $\mathbf{A}_i^m$  the set of components  $C \in \mathbf{A}_i$ , such that the maximal chain of applications of composition operators in the construction of  $C$  has the length  $m$ . In particular,  $\mathbf{A}_i^0 = \mathbf{C}$ .

The proof is by induction of the structural depth of the components involved. The induction hypothesis is the following: *with the restriction of the last quantification to  $\forall C_1^i, \dots, C_n^i \in \mathbf{A}_i^m$ , (4) implies (1) and*

$$\forall C \in \mathbf{A}_2^m, \exists C' \in \mathbf{A}_1^m : \sigma_1(C') = \sigma_2(C). \quad (6)$$

The induction step will consist in proving that if this statement holds for all  $m' < m$ , then it also holds for  $m$ . The base case is  $m = 0$ , i.e. the quantification is over atomic components only.

Consider a pair of operators  $o$  and  $\tilde{o}$ , which satisfy (4) with the last quantification being replaced by  $\forall C_1^i, \dots, C_n^i \in \mathbf{C}$ , and two sets of atomic components  $C_1^i, \dots, C_n^i \in \mathbf{C}$  (for  $i = 1, 2$ ), such that  $\sigma_1(C_k^1) \simeq \sigma_2(C_k^2)$ , for all  $k \in [1, n]$ . Since, by the assumption of the proposition,  $\sigma_1(C_k^2) = \sigma_2(C_k^2)$ , we have by (4)

$$\sigma_1(\tilde{o}\langle C_1^2, \dots, C_n^2 \rangle) = \sigma_2(o\langle C_1^2, \dots, C_n^2 \rangle).$$

Since  $\sigma_1(C_k^1) \simeq \sigma_2(C_k^2) = \sigma_1(C_k^2)$ , by compositionality of  $F_1$ , we have

$$\sigma_1(\tilde{o}\langle C_1^1, \dots, C_n^1 \rangle) \simeq \sigma_1(\tilde{o}\langle C_1^2, \dots, C_n^2 \rangle)$$

and, combining the two,

$$\sigma_1(\tilde{o}\langle C_1^1, \dots, C_n^1 \rangle) \simeq \sigma_2(o\langle C_1^2, \dots, C_n^2 \rangle).$$

Notice that, for  $m = 0$ , (6) holds trivially by the assumption of the proposition with  $C_2 = C_1$ .

Let us now prove the induction step. First of all, consider a component  $C \in \mathbf{A}_2^m$ . Since  $m > 0$ , we have  $C = o\langle C_1, \dots, C_l \rangle$ , for some  $o \in \mathbf{G}_2$  and  $C_1, \dots, C_l \in \mathbf{A}_2^{m-1}$ . Hence, by the induction hypothesis, there exist  $C_1', \dots, C_l' \in \mathbf{A}_1^{m-1}$ , such that  $\sigma_1(C_k') = \sigma_2(C_k)$ , for all  $k \in [1, l]$ . By (4), there exists  $\tilde{o} \in \mathbf{G}_1$ , such that  $\sigma_1(\tilde{o}\langle C_1', \dots, C_l' \rangle) = \sigma_2(o\langle C_1, \dots, C_l \rangle)$ .

$\dots, C_l') = \sigma_2(o\langle C_1, \dots, C_l \rangle) = \sigma_2(C)$ . Denoting  $C' = \tilde{o}\langle C_1', \dots, C_l' \rangle \in \mathbf{A}_1^m$ , we obtain the proof of the induction step for (6).

Consider now a pair of operators  $o$  and  $\tilde{o}$ , which satisfy (4) with the last quantification replaced by  $\forall C_1^i, \dots, C_n^i \in \mathbf{A}_i^m$ . Consider, furthermore, two sets of components  $C_1^i, \dots, C_n^i \in \mathbf{A}_i^m$ , such that  $\sigma_1(C_k^1) \simeq \sigma_2(C_k^2)$ , for all  $k \in [1, n]$ . By (6) (as proven above), there exist  $C_1^{2'}, \dots, C_n^{2'} \in \mathbf{A}_1^m$ , such that  $\sigma_1(C_k^{2'}) = \sigma_2(C_k^2)$ , for all  $k \in [1, n]$ . By (4), we have

$$\sigma_1(\tilde{o}\langle C_1^{2'}, \dots, C_n^{2'} \rangle) = \sigma_2(o\langle C_1^2, \dots, C_n^2 \rangle).$$

Since  $\sigma_1(C_k^1) \simeq \sigma_2(C_k^2) = \sigma_1(C_k^{2'})$ , by compositionality of  $F_1$ , we have

$$\sigma_1(\tilde{o}\langle C_1^1, \dots, C_n^1 \rangle) \simeq \sigma_1(\tilde{o}\langle C_1^{2'}, \dots, C_n^{2'} \rangle)$$

and, combining the two,

$$\sigma_1(\tilde{o}\langle C_1^1, \dots, C_n^1 \rangle) \simeq \sigma_2(o\langle C_1^2, \dots, C_n^2 \rangle),$$

which proves the induction step for (1) and thereby concludes the proof of the proposition.  $\square$

All the frameworks considered in the subsequent sections have structural semantics and follow SOS formats that preserve bisimilarity. Since we consider a bisimilarity-based equivalence relation on the behaviour type, all these frameworks are compositional by Lemma 3. Furthermore, all these frameworks have the same set of atomic components and, up to a canonical extension (see Section 3.3.3), the same semantic domain. They satisfy all the assumptions of Proposition 3, which means that we can prove the positive results about their relative expressiveness by studying the defining mappings of the matching composition operators and showing that their application preserves the equality of behaviours. (Negative results are proven by counterexamples.)

### 3 The BIP component-based framework

#### 3.1 Classical BIP

In this section, we briefly recall BIP and its classical operational semantics, as initially published in [10].

##### 3.1.1 Semantic domain

The behaviour type in BIP is the set of *Labelled Transition Systems* (LTS).

**Definition 8** A *labelled transition system* (LTS) is a triple  $(Q, P, \rightarrow)$ , where  $Q$  is a set of *states*,  $P$  is a set of *ports*, and  $\rightarrow \subseteq Q \times 2^P \times Q$  is a set of *transitions* labelled by sets of ports, such that only self-loops can be labelled by the empty set of ports, i.e.  $(q, \emptyset, q') \in \rightarrow$  implies  $q = q'$ . For  $q, q' \in Q$  and  $a \in 2^P$ , we write  $q \xrightarrow{a} q'$  iff  $(q, a, q') \in \rightarrow$ . A label  $a \in 2^P$  is *active* in a state  $q \in Q$  (denoted  $q \xrightarrow{a}$ ), iff there exists  $q' \in Q$  such that  $q \xrightarrow{a} q'$ . We abbreviate  $q \not\xrightarrow{a} \stackrel{\text{def}}{=} \neg(q \xrightarrow{a})$ .

*Note 1* In the rest of the paper, whenever we speak of a set of LTSs  $B_i = (Q_i, P_i, \rightarrow_i)$ , for  $i \in [1, n]$ , we assume that all  $P_i$  are pairwise disjoint, i.e.  $i \neq j$  implies  $P_i \cap P_j = \emptyset$ . We denote  $P \stackrel{\text{def}}{=} \bigcup_{i=1}^n P_i$ . When the indices are clear from the context, we drop them on transition relations and simply write  $\rightarrow$ .

The equivalence of LTS is defined through a bisimulation relation [32].

**Definition 9** Let  $B_1 = (Q_1, P, \rightarrow_1)$  and  $B_2 = (Q_2, P, \rightarrow_2)$  be two LTS, and let  $R \subseteq Q_1 \times Q_2$  be a binary relation.

- $R$  is a *simulation* iff, for all  $q_1 R q_2$ ,  $q_1 \xrightarrow{a}_1 q'_1$  implies  $q_2 \xrightarrow{a}_2 q'_2$  for some  $q'_2 \in Q_2$  such that  $q'_1 R q'_2$ .
- $R$  is a *bisimulation* iff both  $R$  and  $R^{-1}$  are simulations.

We say that  $B_1$  and  $B_2$  are bisimilar if there exists a bisimulation relation total on both  $Q_1$  and  $Q_2$ .

**Definition 10** Two behaviours  $B_i = (Q_i, P_i, \rightarrow)$ , for  $i = 1, 2$  are *equivalent* if  $P_1 = P_2$ , and the two LTS are bisimilar.

### 3.1.2 Glue operators

BIP glues consist of two layers. *Interaction models* define the sets of allowed *interactions*, i.e. synchronisations between the transitions of their operand components. *Priority models* define the scheduling—or conflict resolution—policies, reducing non-determinism when several synchronisations allowed by the interaction model are enabled simultaneously.

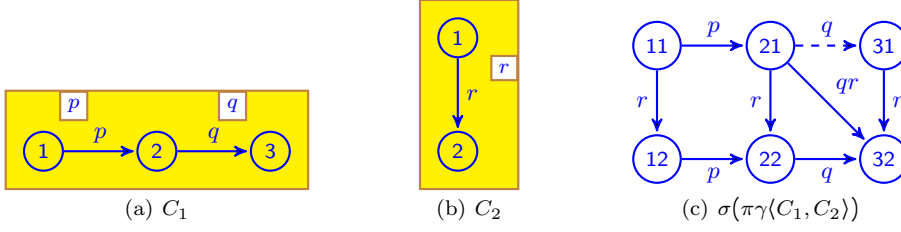
*Interaction models* For given disjoint sets of ports  $P_i$ , for  $i \in [1, n]$ , we denote  $P = \bigcup_{i=1}^n P_i$ . An *interaction model* is a set of interactions  $\gamma \subseteq 2^P$ . The semantics of the application of an interaction model  $\gamma$  is defined for any set of components  $C_1, \dots, C_n$  such that  $\sigma(C_i) = (Q_i, P_i, \rightarrow_i)$ , for  $i \in [1, n]$ , by putting  $\sigma(\gamma(C_1, \dots, C_n)) \stackrel{\text{def}}{=} (Q, P, \rightarrow_\gamma)$ , with  $Q = \prod_{i=1}^n Q_i$  and the minimal transition relation  $\rightarrow_\gamma$  satisfying the rule

$$\frac{a \in \gamma \quad \left\{ q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I \right\} \quad \left\{ q_i = q'_i \mid i \notin I \right\}}{q_1 \dots q_n \xrightarrow{a}_\gamma q'_1 \dots q'_n}, \quad (7)$$

where  $I = \{i \in [1, n] \mid a \cap P_i \neq \emptyset\}$ . Intuitively, this means that an interaction  $a$  allowed by the interaction model  $\gamma$  can be fired when all the components involved in  $a$  are ready to fire the corresponding transitions. All the components that are not involved in  $a$  remain in their current states.

*Priority models* For a component  $C$ , such that  $\sigma(C) = (Q, P, \rightarrow)$ , a *priority model* is a strict<sup>3</sup> partial order  $\pi \subseteq 2^P \times (2^P \setminus \{\emptyset\})$  (we write  $a < b$  as a shorthand for  $(a, b) \in \pi$ ). The semantics of the application of a priority model  $\pi$  is defined by

<sup>3</sup> As opposed to a (non-strict) partial order, which is a reflexive, antisymmetric and transitive relation, a *strict* partial order is an irreflexive and transitive (hence also antisymmetric) one.



**Fig. 2** Components and behaviour for Example 5

putting  $\sigma(\pi\langle C \rangle) \stackrel{\text{def}}{=} (Q, P, \rightarrow_\pi)$ , with the minimal transition relation  $\rightarrow_\pi$  satisfying the rule

$$\frac{q \xrightarrow{a} q' \quad \left\{ q \xrightarrow{b} \mid a < b \right\}}{q \xrightarrow{a}_\pi q'} \quad (8)$$

Intuitively, this means that an interaction can be fired only if no higher-priority interaction is enabled. In this context, it is important to notice that only interactions belonging to the interaction model of a BIP glue operator can be used in the priority model.

**Definition 11** An  $n$ -ary BIP glue operator is a triple  $((P_i)_{i=1}^n, \gamma, \pi)$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and, denoting  $P \stackrel{\text{def}}{=} \bigcup_{i=1}^n P_i$ , the remaining two elements  $\gamma \subseteq 2^P$  and  $\pi \subseteq \gamma \times (\gamma \setminus \{\emptyset\})$  are, respectively, interaction and priority models on  $P$ .

In the remainder of the paper, we omit the sets of ports  $(P_i)_{i=1}^n$  when they are clear from the context.

We will denote by CBIP the framework combining the above defined elements, with LTS being both the behaviour type and the set of atomic components, bisimilarity used as the semantic equivalence and composition operators and the semantic mapping defined as in Definition 11 and rules (7) and (8).

To simplify the notation, we denote the component obtained by applying the glue operator  $((P_i)_{i=1}^n, \gamma, \pi)$  to sub-components  $B_1, \dots, B_n$ , by  $\pi\gamma\langle B_1, \dots, B_n \rangle$  instead of  $((P_i)_{i=1}^n, \gamma, \pi)\langle B_1, \dots, B_n \rangle$ . Furthermore, when  $\pi = \emptyset$ , we write directly  $\gamma\langle B_1, \dots, B_n \rangle$ , omitting  $\pi$ .

*Example 5* Consider the two components  $C_1$  and  $C_2$  shown in Figures 2(a) and 2(b), respectively. We have  $P_1 = \{p, q\}$  and  $P_2 = \{r\}$ , and put  $\gamma = \{p, q, r, qr\}$  and  $\pi = \{q < r\}$ .<sup>4</sup> The semantics of the glue operator defined by the combination of the interaction model  $\gamma$  and the priority model  $\pi$  is given by the following four rules, obtained by composing rules of forms (7) and (8) and removing premises whereof satisfaction does not depend on the state of the operand behaviours (e.g. the premise  $a \in \gamma$  is satisfied in all states):

$$\frac{q_1 \xrightarrow{p} q'_1}{q_1 q_2 \xrightarrow{p} q'_1 q_2}, \quad \frac{q_2 \xrightarrow{r} q'_2}{q_1 q_2 \xrightarrow{r} q_1 q'_2}, \quad \frac{q_1 \xrightarrow{q} q'_1 \quad q_2 \xrightarrow{r} q'_2}{q_1 q_2 \xrightarrow{qr} q'_1 q'_2}, \quad \frac{q_1 \xrightarrow{q} q'_1 \quad q_2 \xrightarrow{r} q'_2}{q_1 q_2 \xrightarrow{q} q'_1 q_2}. \quad (9)$$

<sup>4</sup> To simplify the notation we use the juxtaposition  $\gamma = \{p, q, r, qr\}$  instead of the set notation  $\gamma = \{\{p\}, \{q\}, \{r\}, \{q, r\}\}$  for interactions. Similarly, we directly write  $\pi = \{q < r\}$  instead of  $\pi = \{(q, r)\}$

The behaviour of the composed component  $\pi\gamma\langle B_1, B_2 \rangle$  is shown in Figure 2(c). The dashed arrow  $21 \xrightarrow{q} 31$  shows the transition present only in  $\sigma(\gamma\langle B_1, B_2 \rangle)$ , but not in  $\sigma(\pi\gamma\langle B_1, B_2 \rangle)$ . Solid arrows show the transitions of  $\sigma(\pi\gamma\langle B_1, B_2 \rangle)$ .

Among the transitions labelled by  $q$ , only the transition  $22 \xrightarrow{q} 32$  is enabled and not  $21 \xrightarrow{q} 31$  (Figure 2(c)). Indeed, the negative premise in the fourth rule of (9), generated by the priority  $q < r$ , suppresses the interaction  $q$  when a transition labelled  $r$  is possible in the second component.  $\square$

Let us now recall an important property of the BIP glue operators with the above semantics, which was originally shown in [24]: application of a priority model does not introduce deadlocks.

**Definition 12** Let  $B = (Q, P, \rightarrow)$  be a behaviour. A state  $q \in Q$  is a *deadlock* iff holds the statement  $\forall a \subseteq P, q \not\xrightarrow{a}$ .

**Lemma 4** ([24]) Let  $C_i$ , such that  $\sigma(C_i) = (Q_i, P_i, \rightarrow)$ , for  $i \in [1, n]$ , be a set of components,  $\gamma$  and  $\pi$  be respectively interaction and priority models on  $P = \bigcup_{i=1}^n P_i$ . A state  $q \in \prod_{i=1}^n Q_i$  is a deadlock in  $\sigma(\pi\gamma\langle C_1, \dots, C_n \rangle)$  if and only if it is a deadlock in  $\sigma(\gamma\langle C_1, \dots, C_n \rangle)$ .

*Proof.* The “if” implication is trivial. To prove the “only if” implication, assume that, for some  $a \in \gamma$ , we have  $q \xrightarrow{a} \gamma$ . Let  $b \subseteq P$  be an interaction, maximal w.r.t.  $\pi$ , such that  $b \in \gamma$ ,  $a < b$  and  $q \xrightarrow{b} \gamma$ . If such  $b$  exists, holds  $q \xrightarrow{b} \pi$ . Otherwise holds  $q \xrightarrow{a} \pi$ . In both cases,  $q$  is not a deadlock in  $\sigma(\pi\gamma\langle C_1, \dots, C_n \rangle)$ .  $\square$

Notice that this proof does not rely on  $\pi$  being a strict partial order. The lemma can be generalised to any *acyclic* relation  $\pi \subseteq \gamma \times \gamma$ .

### 3.1.3 BIP-like SOS format

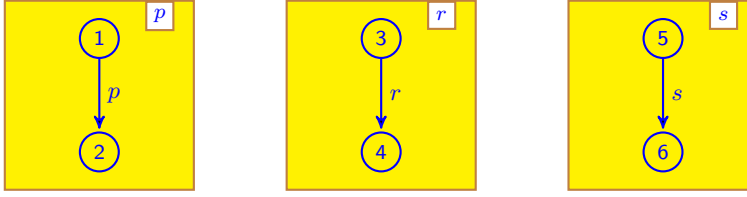
Observe that the rules in (9) are obtained by composing rules of forms (7) and (8). In particular, the fourth rule is obtained by the following derivation:

$$\frac{\frac{q \in \gamma \quad q_1 \xrightarrow{q} q_1' \quad q_2 = q_2'}{q_1 q_2 \xrightarrow{q} q_1' q_2'} \quad \frac{r \notin \gamma \quad \vee \quad q_2 \xrightarrow{r} \gamma}{q_1 q_2 \xrightarrow{r} \gamma} (*)}{q_1 q_2 \xrightarrow{q} \pi \quad q_1' q_2'} . \quad (10)$$

The sub-derivation (\*) in (10) is obtained by negating the premises of the instance of (7) with  $a = r$ . This is possible because the transition relation in  $\sigma(\gamma\langle B_1, B_2 \rangle)$  is defined by (7) inductively, i.e. it is the minimal transition relation satisfying (7).

In (9), we have simplified (10) by removing premises, whereof satisfaction does not depend on the state of the operand behaviours:  $q \in \gamma$  (satisfied in all states) and  $r \notin \gamma$  (dissatisfied in all states), and by replacing  $q_2'$  with  $q_2$ . Notice that the priority  $q < r$  affects the behaviour of the composed system only because  $r \in \gamma$ . Indeed, if  $r$  did not belong to  $\gamma$ , the premise  $r \notin \gamma$  would always be satisfied independently of the state of the system.

Every BIP glue operator is a combination of a (possibly trivial) interaction model with a (possibly trivial) priority model. By merging a rule of form (8)



**Fig. 3** Components for Example 6

with a layer consisting of one or several rules of form (7)—as in the example (10) above—and simplifying by removing the constant premises, we always obtain a rule in the format

$$\frac{\left\{ q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I \right\} \quad \left\{ q_i = q'_i \mid i \notin I \right\} \quad \left\{ q_j \xrightarrow{b_j} (j, b_j) \in H \right\}}{q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n}, \quad (11)$$

where  $I = \{i \in [1, n] \mid a \cap P_i \neq \emptyset\}$ , whereas  $H \subseteq [1, n] \times 2^P$  and  $b_j \in 2^P \setminus \{\emptyset\}$ , for each  $(j, b_j) \in H$ . Below we call this format *BIP-like SOS*. Thus, the semantics of every BIP glue operator can be defined by a set of rules in the BIP-like SOS format.

*Note 2* Notice that for two rules in *any given format*, if their conclusions coincide but the set of all the premises of one of the rules is contained in that of the other rule, then the latter rule is *redundant*. Indeed, whenever the rule with more premises can be applied, so can be the rule with less premises. Similarly, if a rule has contradictory premises, as for example  $q \xrightarrow{a} q'$  and  $q \not\xrightarrow{a}$ , it is redundant, since it can never be applied.

Below, we will consider only sets of rules that do not contain such redundant ones and will sometimes emphasize this fact by explicitly calling such sets *non-redundant*.

**Definition 13** A *BIP-like SOS operator* is the composition operator defined as  $((P_i)_{i=1}^n, \mathcal{R})$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and  $\mathcal{R}$  is a non-redundant set of BIP-like SOS rules (11).

We will denote by BSOS the framework with the same atomic components and the same semantic domain as CBIP (i.e. LTS and bisimilarity) and with BIP-like SOS glue operators with their semantics defined inductively by the corresponding sets of rules.

The following example shows that BSOS does not possess flattening. Although, in Section 2.2, we have only defined uniform flattening, this example shows that BSOS does not possess even a weaker form defined in [4], where different operators are admitted for different sets of operand components.

*Example 6* Consider two composition operators defined by sets of rules in the BIP-like SOS format:

$$o_1 : \frac{q_1 \xrightarrow{p} q'_1}{q_1 q_2 q_3 \xrightarrow{p} q'_1 q_2 q_3}, \quad \frac{q_2 \xrightarrow{r} q'_2 \quad q_3 \xrightarrow{s} q'_3}{q_1 q_2 q_3 \xrightarrow{r} q_1 q'_2 q_3}, \quad \frac{q_3 \xrightarrow{s} q'_3}{q_1 q_2 q_3 \xrightarrow{s} q_1 q_2 q'_3}. \quad (12)$$



$$o_2 : \frac{q \xrightarrow{p} q' \quad q \xrightarrow{r} q'}{q \xrightarrow{p} q'}, \quad \frac{q \xrightarrow{r} q'}{q \xrightarrow{r} q'}, \quad \frac{q \xrightarrow{s} q'}{q \xrightarrow{s} q'}. \quad (13)$$

Consider three components  $C_1, C_2$  and  $C_3$  shown in Figure 3 and a composed component  $o_2\langle o_1\langle C_1, C_2, C_3 \rangle \rangle$ . Transition  $p$  in  $\sigma(o_2\langle o_1\langle C_1, C_2, C_3 \rangle \rangle)$  is available only in states where  $r$  was not enabled in  $\sigma(o_1\langle C_1, C_2, C_3 \rangle)$ . In particular, in the state 135,  $p$  is available, since  $r$  is blocked by  $s$  in the second rule defining  $o_1$ . To the contrary, in the state 136,  $p$  is blocked by  $r$  in the first rule defining  $o_2$ . The only difference between the active actions in the global states 135 and 136 is that  $s$  is active in 135, but not in 136. Thus, to obtain the same behaviour as in  $o_2\langle o_1\langle C_1, C_2, C_3 \rangle \rangle$  using a single BIP-like SOS operator, we have to test the activity of  $s$ . However, the BIP-like SOS rule format (11) only allows the use of the negative premise  $q_3 \xrightarrow{s} \neg$ , which would induce the opposite behaviour, i.e. blocking  $p$  in 135 instead of 136. Thus, it is impossible to represent the behaviour of  $o_2\langle o_1\langle C_1, C_2, C_3 \rangle \rangle$  with a single BIP-like SOS composition operator.

In Section 4.1, we will show that CBIP does not have even weak full expressiveness w.r.t. BSOS. In the following subsections, we first introduce two useful variations of the BSOS format (Section 3.2), then we present some BIP modifications that we use in our study (Section 3.3).

### 3.2 Variations of the BIP-like SOS format

#### 3.2.1 Witness BIP-like SOS

In Example 6 above, a transition labelled by  $p$  is available in all states of the composed component, where  $r$  is not enabled. The enabledness of  $r$  is controlled by the operator  $o_1$  as follows:  $r$  is disabled in all states of  $\sigma(o_1\langle C_1, C_2, C_3 \rangle)$ , where at least one premise of the corresponding rule in  $o_1$  is not satisfied. Thus, in order to flatten the composition  $o_2 \circ o_1$ , the rule for the transition  $p$  should be split into two: one with the premise  $q_2 \xrightarrow{r} \neg$  that allows  $p$  in all states where  $r$  is not enabled in the behaviour  $B_2$ ; and another with the premise  $\neg q_3 \xrightarrow{s} \neg$  or, equivalently,  $q_3 \xrightarrow{s}$ . This latter rule would allow  $p$  in all states, where  $s$  is enabled; in such states  $r$  is blocked by the operator  $o_1$ . We call premises of the form  $q_3 \xrightarrow{s}$  *witness*.

**Definition 14** A *Witness BIP-like SOS operator* is a composition operator defined as  $((P_i)_{i=1}^n, \mathcal{R})$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and  $\mathcal{R}$  is a non-redundant set of SOS rules in the following format:

$$\frac{\left\{ q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I \right\} \quad \{ q_i = q'_i \mid i \notin I \} \quad \left\{ q_j \xrightarrow{b_j} \mid (j, b_j) \in H \right\} \quad \{ q_k \xrightarrow{c_k} \mid (k, c_k) \in L \}}{q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n}, \quad (14)$$

with  $I = \{i \in [1, n] \mid a \cap P_i \neq \emptyset\}$  and  $H, L \subseteq [1, n] \times 2^P$ , such that, for each  $(j, b_j) \in H$ , holds  $b_j \in 2^{P_j}$  and, for each  $(k, c_k) \in L$ , holds  $c_k \in 2^{P_k}$ .

We call the premises of the form  $q_i \xrightarrow{a \cap P_i} q_i'$  *firing*; the premises of the form  $q_j \xrightarrow{b_j} q_j'$  *negative*; and the premises of the form  $q_k \xrightarrow{c_k} q_k'$  *witness*.

We will denote by WBSOS the framework, which has the same semantic domain and atomic components as above, but uses witness BIP-like SOS operators with the semantic mapping defined by (14).

**Proposition 4** *WBSOS possesses uniform flattening.*

*Sketch of the proof.* The proof follows by taking the classical composition of SOS rules. We only have to prove that this composition preserves the format (14). This is straightforward for the firing premises. For the witness premises, the only difference is that the target state is irrelevant, so the top-level firing premises in the composition of two rules become witness premises by dropping the target state. For negative premises, top-level firing and witness premises become negative in the composed rule, while top-level negative rules become witness. Thus the format is, indeed, preserved.  $\square$

*Example 7* To illustrate the above proof, consider the following three rules:

$$\frac{q_1 \xrightarrow{a} q_1' \quad q_{234} \xrightarrow{bc} q_{234}' \quad q_{56} \xrightarrow{d} q_{56}'}{q_1 q_{234} q_{56} \xrightarrow{a} q_1' q_{234}' q_{56}'}, \quad \frac{q_2 \xrightarrow{b} q_2' \quad q_3 \xrightarrow{e} q_3' \quad q_4 \xrightarrow{c} q_4'}{q_2 q_3 q_4 \xrightarrow{bc} q_2' q_3' q_4'}, \quad \frac{q_5 \xrightarrow{d} q_5' \quad q_6 \xrightarrow{f} q_6'}{q_5 q_6 \xrightarrow{d} q_5' q_6'}.$$

Substituting the second and third rules for the corresponding premises of the first rule, we obtain the following three rules, differing only in their second premises:

$$\frac{q_1 \xrightarrow{a} q_1' \quad q_2 \xrightarrow{b} q_2' \quad q_5 \xrightarrow{d} q_5' \quad q_6 \xrightarrow{f} q_6'}{q_1 q_2 q_3 q_4 q_5 q_6 \xrightarrow{a} q_1' q_2' q_3 q_4 q_5' q_6'}, \quad \frac{q_1 \xrightarrow{a} q_1' \quad q_3 \xrightarrow{e} q_3' \quad q_5 \xrightarrow{d} q_5' \quad q_6 \xrightarrow{f} q_6'}{q_1 q_2 q_3 q_4 q_5 q_6 \xrightarrow{a} q_1' q_2 q_3' q_4 q_5' q_6'},$$

$$\frac{q_1 \xrightarrow{a} q_1' \quad q_4 \xrightarrow{c} q_4' \quad q_5 \xrightarrow{d} q_5' \quad q_6 \xrightarrow{f} q_6'}{q_1 q_2 q_3 q_4 q_5 q_6 \xrightarrow{a} q_1' q_2 q_3 q_4' q_5' q_6'}.$$

Indeed, all these rules respect the format (14).

### 3.2.2 Inhibiting relation

In this section, we provide a technical definition that will be instrumental in the rest of the paper.

**Definition 15** We call a function  $f : X \rightarrow Y$  a *choice function* and write

$$f : X \rightrightarrows Y : (x). \Phi(x, f),$$

where  $\Phi(x, y)$  is a predicate over  $X \times Y$ , if, for each  $x \in X$ , holds  $\Phi(x, f(x))$ .

For example, for a set  $\mathcal{R}$  of rules in the format (11) above, the choice function

$$h : \mathcal{R} \rightrightarrows \bigcup_{r \in \mathcal{R}} H^r : (r). h \in H^r \quad (15)$$

selects, for each rule  $r \in \mathcal{R}$ , one of its negative premises. (It is well defined for a set of rules  $\mathcal{R}$ , where each rule has at least one negative premise, i.e.  $H^r \neq \emptyset$ .)

Consider a family of disjoint sets of ports  $(P_i)_{i=1}^n$  and a non-redundant set  $\mathcal{R}$  of rules in format (11):

$$r \in \mathcal{R} : \frac{\left\{ q_i \xrightarrow{a^r \cap P_i} q_i' \mid i \in I^r \right\} \quad \left\{ q_i = q_i' \mid i \notin I^r \right\} \quad \left\{ q_j \xrightarrow{b_j^r} (j, b_j^r) \in H^r \right\}}{q_1 \dots q_n \xrightarrow{a} q_1' \dots q_n'}$$

Given an interaction  $a$ , we denote  $R_a = \{r \in \mathcal{R} \mid a^r = a \wedge H^r \neq \emptyset\}$ , the set of rules with the conclusion labelled by  $a$ , which have negative premises. Notice that, since  $\mathcal{R}$  is non-redundant, there are two possibilities: either 1) there is exactly one rule in  $\mathcal{R}$  with conclusion  $a$  and no negative premises, in which case  $R_a = \emptyset$ ; or 2) all rules with conclusion  $a$  have negative premises, in which case  $R_a = \{r \in \mathcal{R} \mid a^r = a\}$ .

Clearly, for the interaction  $a$  to be inhibited by the negative premises, one negative premise must be involved for each rule in  $R_a$ .

We will now define the *inhibiting relation*  $\pi \subseteq 2^P \times (2^{2^P \setminus \{\emptyset\}} \setminus \emptyset)$  by defining  $\pi(a)$  for each interaction  $a$  among the conclusions of the rules in  $\mathcal{R}$ , such that  $R_a \neq \emptyset$ . Intuitively, the set  $\pi(a)$  comprises all possible sets of interactions, formed by combining negative premises from all the rules with conclusion  $a$ , that would block  $a$  if enabled simultaneously. We need  $\pi(a)$  to be a set of sets of interactions, rather than a set of interactions, in order to distinguish, for example, two transitions  $q_j \xrightarrow{b_j^{r_1}}$  and  $q_j \xrightarrow{b_j^{r_2}}$  (with the same  $j$ ) from a single one  $q_j \xrightarrow{b_j^{r_1} b_j^{r_2}}$ .

Fix an interaction  $a$  as above and consider the choice function  $h$  in (15) over  $R_a$ , i.e.  $h : R_a \rightrightarrows \bigcup_{r \in R_a} H^r : (r).h \in H^r$ . We denote

$$J_h \stackrel{\text{def}}{=} \{j \in [1, n] \mid \exists r \in R_a : h(r) = (j, -)\} \quad (16)$$

the set of indices involved in the premises chosen by  $h$ . Notice that different premises can be associated to the same index in  $J_h$ . Therefore, for a given choice  $h$ , we consider another choice function that selects, for each index in  $J_h$ , the label of one premise among those chosen by  $h$  (recall (11) that  $H^r \subseteq [1, n] \times 2^P$ , for each  $r \in R$ ):

$$b_h : J_h \rightrightarrows 2^P : (j).(j, b_h) \in h(R_a). \quad (17)$$

Consider now the interaction  $b = \bigcup_{j \in J_h} b_h(j)$ . By construction of  $h$ ,  $J_h$  and  $b_h$ , we have  $b \cap P_j = b_h(j)$  (recall (11) that, for each  $(j, b_j) \in H^r$ , we have  $b_j \in 2^{P_j} \setminus \{\emptyset\}$ ). Hence, whenever  $b$  is enabled, we have  $q_j \xrightarrow{b_h(j)}$ , i.e. all the premises involved in the construction of  $b$  are dissatisfied. Thus, by combining all possible choices of  $b_h$  we obtain a set of interactions that, if all are enabled, guarantees that each of the rules with the conclusion labelled by  $a$  has a dissatisfied premise, thereby effectively blocking  $a$ . In order to define  $\pi(a)$ , we consider all possible choices of  $h$  to form a set of sets of interactions (in (18) below, we reproduce the

definitions from (15), (16) and (17)):

$$\pi(a) \stackrel{\text{def}}{=} \left\{ \left\{ \bigcup_{j \in J_h} b_h(j) \mid b_h : J_h \Rightarrow 2^P : (j).(j, b_h) \in h(R_a) \right\} \mid \begin{array}{l} h : \mathcal{R} \Rightarrow \bigcup_{r \in \mathcal{R}} H^r : (r).h \in H^r, \\ J_h = \{j \in [1, n] \mid \exists r \in R_a : h(r) = (j, -)\} \end{array} \right\}. \quad (18)$$

To illustrate the need for  $\pi(a)$  to be a set of sets of ports, compare the following four sets of rules:<sup>5</sup>

$$\begin{array}{ll} \begin{array}{l} P_1 = \{p\}, \\ P_2 = \{r, s\} \end{array} & \frac{q_1 \xrightarrow{p} q_1' \quad q_2 \xrightarrow{rs} q_2'}{q_1 q_2 \xrightarrow{p} q_1' q_2'} \quad \pi = \{(p, rs)\} \\ \\ \begin{array}{l} P_1 = \{p\}, P_2 = \{r\}, \\ P_3 = \{s\} \end{array} & \frac{q_1 \xrightarrow{p} q_1' \quad q_2 \xrightarrow{r} q_2'}{q_1 q_2 q_3 \xrightarrow{p} q_1' q_2' q_3}, \frac{q_1 \xrightarrow{p} q_1' \quad q_3 \xrightarrow{s} q_3'}{q_1 q_2 q_3 \xrightarrow{p} q_1' q_2 q_3'} \quad \pi = \{(p, rs)\} \\ \\ \begin{array}{l} P_1 = \{p\}, \\ P_2 = \{r, s\} \end{array} & \frac{q_1 \xrightarrow{p} q_1' \quad q_2 \xrightarrow{r} q_2'}{q_1 q_2 \xrightarrow{p} q_1' q_2'}, \frac{q_1 \xrightarrow{p} q_1' \quad q_2 \xrightarrow{s} q_2'}{q_1 q_2 \xrightarrow{p} q_1' q_2'} \quad \pi = \{(p, r \cdot s)\} \\ \\ \begin{array}{l} P_1 = \{p\}, P_2 = \{r\}, \\ P_3 = \{s\} \end{array} & \frac{q_1 \xrightarrow{p} q_1' \quad q_2 \xrightarrow{r} q_2' \quad q_3 \xrightarrow{s} q_3'}{q_1 q_2 q_3 \xrightarrow{p} q_1' q_2' q_3'} \quad \pi = \{(p, r), (p, s)\} \end{array}$$

**Definition 16** Given a family of disjoint sets of ports  $(P_i)_{i=1}^n$  and a non-redundant set  $\mathcal{R}$  of rules in format (11), the corresponding *inhibiting relation*  $\pi$  is defined by (18). If, for each  $a$  in the domain of  $\pi$ , all sets in  $\pi(a)$  are singleton, we say that the inhibiting relation  $\pi$  is *simple*. We call *complex* inhibiting relations that are not simple.

Among the four examples above, only the third one is *not* simple. In the first and second examples, the set  $\pi(p) = \{\{rs\}\}$  contains one singleton set comprising the interaction  $rs$ . In the third example,  $\pi(p) = \{\{r, s\}\}$  contains one set comprising two singleton interactions  $r$  and  $s$ . Finally, in the fourth example,  $\pi(p) = \{\{r\}, \{s\}\}$  contains two singleton sets comprising singleton interactions  $r$  and  $s$ , respectively.

### 3.2.3 Simple and acyclic restrictions of BIP-like SOS

For the sake of rendering the expressiveness hierarchy more explicit, we consider the two restrictions of BSOS below.

First consider the operator defined by the rules

$$\begin{array}{c} \frac{q_1 \xrightarrow{p} q_1' \quad q_2 \xrightarrow{r} q_2' \quad q_3 \xrightarrow{s} q_3'}{q_1 q_2 q_3 \xrightarrow{p} q_1' q_2' q_3'}, \quad \frac{q_2 \xrightarrow{r} q_2' \quad q_3 \xrightarrow{s} q_3'}{q_1 q_2 q_3 \xrightarrow{r} q_1 q_2' q_3'}, \\ \frac{q_2 \xrightarrow{r} q_2' \quad q_1 \xrightarrow{p} q_1'}{q_1 q_2 q_3 \xrightarrow{r} q_1' q_2' q_3}, \quad \frac{q_3 \xrightarrow{s} q_3' \quad q_1 \xrightarrow{p} q_1'}{q_1 q_2 q_3 \xrightarrow{s} q_1 q_2 q_3'}. \quad (19) \end{array}$$

<sup>5</sup> We denote by  $r \cdot s$  the set of two singleton interactions  $r$  and  $s$ , as opposed to  $rs$ , which denotes one interaction consisting of the two ports.

It has the simple inhibiting relation  $\pi = \{(p, r), (p, s), (r, ps), (s, p)\}$ . Notice that the (singleton) interactions  $p$  and  $s$  form a cycle in the inhibiting relation and they would block each other if enabled simultaneously.

Since all sets in the codomain of a simple inhibiting relation are singleton, they can be systematically opened, e.g. replacing  $(a, \{b\})$  by  $(a, b)$ , without loss of information. This implies that a simple inhibiting relation on  $2^P \times (2^{2^P \setminus \{\emptyset\}} \setminus \emptyset)$  can be equivalently considered as a relation on  $2^P \times (2^P \setminus \{\emptyset\})$ .

**Definition 17** We say that a simple inhibiting relation is *acyclic* if it does not have any cycles when considered as a relation on  $2^P \times (2^P \setminus \{\emptyset\})$ .

**Definition 18** A *simple BIP-like SOS operator* is a composition operator defined as  $((P_i)_{i=1}^n, \mathcal{R})$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and  $\mathcal{R}$  is a non-redundant set of BIP-like SOS rules (11) with a simple inhibiting relation.

**Definition 19** An *acyclic BIP-like SOS operator* is a composition operator defined as  $((P_i)_{i=1}^n, \mathcal{R})$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and  $\mathcal{R}$  is a non-redundant set of BIP-like SOS rules (11) with a simple and acyclic inhibiting relation.

We will denote by SiBSOS and AcBSOS the frameworks, which have the same semantic domain and atomic components as above, but use, respectively, simple and acyclic BIP-like SOS operators with the semantic mapping defined by the corresponding sets of rules in format (11).

*Note 3* Notice that, for any classical BIP operator  $((P_i)_{i=1}^n, \gamma, \pi)$ , the inhibiting relation of the set of BIP-like SOS rules defining its semantics coincides with  $\pi$  and, therefore, is simple and acyclic.

### 3.3 Modifications of BIP

#### 3.3.1 Relaxation of priority models

The first option proposed in [4] is the relaxation of the priority model.

**Definition 20** Let  $P$  be a set of ports. A *relaxed priority model* on  $P$  is a relation  $\pi \subseteq 2^P \times (2^P \setminus \{\emptyset\})$ .

The key point that distinguishes relaxed priority models from the classical ones is that  $\pi$  can be an arbitrary relation.

**Definition 21** A *relaxed BIP operator* is a triple  $((P_i)_{i=1}^n, \gamma, \pi)$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and, denoting  $P \stackrel{\text{def}}{=} \bigcup_{i=1}^n P_i$ ,  $\gamma \subseteq 2^P$  is an interaction model and  $\pi \subseteq 2^P \times (2^P \setminus \{\emptyset\})$  is a relaxed priority model.

Notice that we do not require the relation  $\pi$  to be acyclic or transitive. If all interactions involved in a cyclic dependency in  $\pi$  are enabled simultaneously, they block each other, potentially introducing a deadlock. We have also removed the restriction  $\pi \subseteq \gamma \times \gamma$ , which requires a slight modification of semantics. Clearly, the behaviour of  $\gamma \langle C_1, \dots, C_n \rangle$  does not have transitions, whereof labels are not in  $\gamma$ . Hence, the rules defining the semantics of a priority model will have no effect for

all priorities outside  $\gamma \times \gamma$ . Thus, we need to apply interaction and priority models simultaneously.

Given components  $(C_i)_{i=1}^n$ , such that  $\sigma(C_i) = (Q_i, P_i, \rightarrow_i)$ , the semantics of the simultaneous application of an interaction model  $\gamma$  and a priority model  $\pi$  is defined by putting  $\sigma(\pi\gamma(C_1, \dots, C_n)) \stackrel{\text{def}}{=} (Q, P, \rightarrow_{\pi\gamma})$ , with  $Q = \prod_{i=1}^n Q_i$  and the minimal transition relation  $\rightarrow_{\pi\gamma}$  inductively defined by the set of rules

$$\left\{ \frac{\left\{ \begin{array}{l} \{q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I\} \quad \{q_i = q'_i \mid i \notin I\} \\ \{q_{j(b)} \xrightarrow{b \cap P_{j(b)}} \mid b \in \pi(a)\} \end{array} \right.}{q_1 \dots q_n \xrightarrow{a}_{\pi\gamma} q'_1 \dots q'_n} \left| \begin{array}{l} a \in \gamma, \\ j : \pi(a) \Rightarrow [1, n] : (b).b \cap P_j \neq \emptyset \end{array} \right. \right\}, \quad (20)$$

where  $I = \{i \in [1, n] \mid a \cap P_i \neq \emptyset\}$ .

The negative premises of each rule in (20) are determined by the choice function  $j : \pi(a) \Rightarrow [1, n] : (b).b \cap P_j \neq \emptyset$ . Intuitively, for an interaction  $a$  to be blocked by a higher-priority interaction  $b$ , every rule with the conclusion labelled by  $a$  must have at least one of its premises violated. Here, for every such rule and for every higher-priority interaction  $b$ , we choose one component, i.e. that indexed by  $j(b)$ , involved in  $b$  and test that the corresponding sub-interaction is disabled:  $q_{j(b)} \xrightarrow{b \cap P_{j(b)}} \not\rightarrow$ . Thus, whenever  $b$  is enabled, every rule authorising  $a$  has at least one of its premises violated.

We will denote by RBIP the framework that extends CBIP with relaxed operators using the semantic mapping defined by (20). Notice that for the operators with classical priority models the classical and the modified semantics coincide.

### 3.3.2 BIP with complex priority relations

A further modification of BIP consists in extending priority models to rely on sets of interactions to inhibit another one, instead of a single interaction. This extension will allow us to recover the full expressiveness of BIP-like SOS through complex inhibiting relations.

**Definition 22** Let  $P$  be a set of ports. A *complex priority model* on  $P$  is a relation  $\pi \subseteq 2^P \times (2^{2^P \setminus \{\emptyset\}} \setminus \{\emptyset\})$ .

**Definition 23** A *complex BIP operator* is a triple  $((P_i)_{i=1}^n, \gamma, \pi)$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports,  $\gamma \subseteq 2^P$  is an interaction model and  $\pi \subseteq 2^P \times (2^{2^P \setminus \{\emptyset\}} \setminus \{\emptyset\})$  is a complex priority model.

Given components  $(C_i)_{i=1}^n$ , such that  $\sigma(C_i) = (Q_i, P_i, \rightarrow_i)$ , the semantics of the simultaneous application of an interaction model  $\gamma$  and a complex priority model  $\pi$  is defined by putting  $\sigma(\pi\gamma(C_1, \dots, C_n)) \stackrel{\text{def}}{=} (Q, P, \rightarrow_{\pi\gamma})$ , with  $Q = \prod_{i=1}^n Q_i$  and the minimal transition relation  $\rightarrow_{\pi\gamma}$  inductively defined by the set of rules

$$\left\{ \frac{\left\{ \begin{array}{l} \{q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I\} \quad \{q_i = q'_i \mid i \notin I\} \\ \{q_{j(\alpha)} \xrightarrow{b(\alpha) \cap P_{j(\alpha)}} \mid \alpha \in \pi(a)\} \end{array} \right.}{q_1 \dots q_n \xrightarrow{a}_{\pi\gamma} q'_1 \dots q'_n} \left| \begin{array}{l} a \in \gamma, \\ b : \pi(a) \Rightarrow \bigcup \pi(a) : (\alpha).b \in \alpha, \\ j : \pi(a) \Rightarrow [1, n] : (\alpha).b(\alpha) \cap P_j \neq \emptyset \end{array} \right. \right\}, \quad (21)$$

where  $I = \{i \in [1, n] \mid a \cap P_i \neq \emptyset\}$ .

The negative premises of each rule in (21) are determined by two choice functions  $b : \pi(a) \Rightarrow \bigcup \pi(a) : (\alpha).b \in \alpha$  and  $j : \pi(a) \Rightarrow [1, n] : (\alpha).b(\alpha) \cap P_j \neq \emptyset$ . Intuitively, for an interaction  $a$  to be blocked by a higher-priority set of interactions  $\alpha$ , every rule with the conclusion labelled by  $a$  must have at least one of its premises violated. Here, for every such rule and for every higher-priority set of interactions  $\alpha$ , the choice function  $b$  selects the interaction  $b(\alpha) \in \alpha$ , while the choice function  $j$  selects one of the components involved in  $b(\alpha)$ , i.e. such that  $b(\alpha) \cap P_{j(\alpha)} \neq \emptyset$ . The corresponding premise  $q_{j(\alpha)} \xrightarrow{b(\alpha) \cap P_{j(\alpha)}} \text{tests whether } b(\alpha) \text{ is disabled. Thus, whenever all interactions in } \alpha \text{ are enabled, every rule authorising } a \text{ has at least one of its premises violated.}$

We will denote by XBIP the framework which has the same semantic domain and atomic components as above, but uses complex BIP operators with the semantic mapping defined by (21).

Notice that by taking, for a relaxed priority model  $\pi \in 2^P \times (2^P \setminus \emptyset)$ , the complex priority model  $\tilde{\pi} \stackrel{\text{def}}{=} \{(a, \{b\}) \mid (a, b) \in \pi\}$ , relaxed BIP operators can be canonically transformed into complex ones. The relaxed and the complex semantics for these operators coincide, since there is only one possible choice function  $b$  in the semantics (21) of  $\tilde{\pi}$ .

### 3.3.3 Extension of behaviour with the offer predicate

In the previous sections, we have observed that with the classical semantics information about transitions enabled in the components was lost after the application of the interaction model, preventing relaxed priority from taking effect. We have addressed this to some extent by modifying the way the semantics mapping is defined through combining the SOS rules for interaction and for priority models into one.

An alternative approach, consists in extending the definition of the component behaviour to integrate part of this information about the active transitions of its subcomponents. In order to do this we extend the notion of behaviour with an additional *offer* predicate [4, 13]. The notable difference with the approach of the previous section is that information is made available about the active transitions of the atomic subcomponents at the lowest levels of the hierarchy instead of the one immediately underneath the considered one.

**Definition 24** An *extended behaviour* is a quadruple  $B = (Q, P, \rightarrow, \uparrow)$ , where  $(Q, P, \rightarrow)$  is an LTS and  $\uparrow$  is an *offer* predicate on  $Q \times P$ , such that  $q \uparrow p$  holds (the port  $p \in P$  is *offered* in the state  $q \in Q$ ) whenever there is a transition from  $q$  containing  $p$ , that is  $(\exists a \in 2^P : p \in a \wedge q \xrightarrow{a}) \Rightarrow q \uparrow p$ .

The offer predicate extends to sets of ports: for  $a \in 2^P$ ,  $q \uparrow a \stackrel{\text{def}}{=} \bigwedge_{p \in a} q \uparrow p$ . Notice that  $q \uparrow \emptyset \equiv \text{true}$ . We denote  $q \nmid a \stackrel{\text{def}}{=} \neg(q \uparrow a) = \bigvee_{p \in a} q \nmid p$ .

**Definition 25** Two extended behaviours  $B_i = (Q_i, P_i, \rightarrow_i, \uparrow_i)$ , with  $i = 1, 2$ , are *equivalent* if  $P_1 = P_2$  and there exists a bisimulation relation  $R \subseteq Q_1 \times Q_2$ , total on both  $Q_1$  and  $Q_2$ , such that the offer predicates coincide on bisimilar states, i.e. for all  $(q_1, q_2) \in R$  and  $p \in P_1$ , holds  $q_1 \uparrow_1 p \Leftrightarrow q_2 \uparrow_2 p$ .

*Note 4* In the rest of the paper we assume that all behaviours are extended. The classical and relaxed BIP as well as various formats of SOS rules require a simple modification of the corresponding semantic mapping in order to work with extended behaviours. Given a framework  $F = (\mathbf{G}, \mathbf{C}, \mathbf{B}, \approx, \sigma)$ , where the semantic domain  $(\mathbf{B}, \approx)$  is the LTS with bisimilarity-based equivalence from Definition 10, we define the corresponding extended version  $F' = (\mathbf{G}, \mathbf{C}, \mathbf{B}', \approx', \sigma')$ , where  $\mathbf{B}'$  is the set of extended behaviours as in Definition 24,  $\approx'$  is the equivalence from Definition 25 and the semantic mapping  $\sigma'$  is defined by putting  $\sigma'(C) \stackrel{\text{def}}{=} (Q, P, \rightarrow, \uparrow)$ , with  $(Q, P, \rightarrow) = \sigma(C)$  and

$$q \uparrow p \stackrel{\text{def}}{\iff} \begin{cases} \exists a \in 2^P : p \in a \wedge q \xrightarrow{a}, & \text{if } C \in \mathbf{C}, \\ \exists i \in [1, n] : q_i \uparrow p, & \text{if } C = o\langle C_1, \dots, C_n \rangle \text{ and } q = (q_1, \dots, q_n). \end{cases}$$

Notice that the predicate  $\uparrow$  is defined by the same rule, for all composition operators.

In [13], a set of composition operators has been proposed that utilise the information about offered ports. They are defined by the rules in the following format:

$$\frac{\begin{array}{l} \{q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I\} \quad \{q_i = q'_i \mid i \notin I\} \\ \{q_k \nrightarrow b_k \mid (k, b_k) \in H\} \quad \{q_j \uparrow c_j \mid j \in J\} \end{array}}{q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n}, \quad (22)$$

where  $I = \{i \in [1, n] \mid a \cap P_i \neq \emptyset\}$ ,  $J \subseteq [1, n]$ ,  $c_j \in 2^{P_j}$ , for all  $j \in J$ ,  $H \in [1, n] \times 2^P$  and for each  $(k, b_k) \in H$  holds  $b_k \in 2^{P_k} \setminus \{\emptyset\}$ . In (22), there are three types of premises respectively called *firing*, *negative* and *activation* premises. Firing and activation premises are collectively called *positive*. Notice that, while the firing premises are the same as in BSOS, the negative ones are based on a different predicate. Notice, furthermore, that  $q \uparrow c_1 \wedge q \uparrow c_2 = q \uparrow (c_1 \cup c_2)$ . Hence one activation premise per component is sufficient to define any inference rule.

**Definition 26** An *FNASOS operator* is the composition operator defined as  $((P_i)_{i=1}^n, \mathcal{R})$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and, denoting  $P \stackrel{\text{def}}{=} \bigcup_{i=1}^n P_i$ ,  $\mathcal{R}$  is a non-redundant set of SOS rules in the format (22).

We denote by FNASOS the framework with the same atomic components as in CBIP, the semantic domain consisting of the set of extended behaviours and their equivalence as per Definition 24 and Definition 25, and the set of FNASOS composition operators with the semantic mapping defined by the corresponding sets of rules (22).

**Proposition 5** *FNASOS possesses uniform flattening.*

*Sketch of the proof.* The proof follows by taking the classical composition of SOS rules. Notice that, for any  $n$ -ary composition operator, the definition of the offer predicate can also be written as a set of SOS rules:

$$\left\{ \frac{q_i \uparrow p}{q_1 \dots q_n \uparrow p} \mid i \in [1, n] \right\}. \quad (23)$$



Thus, we only have to notice that this composition trivially preserves the format (22). Indeed, for firing premises, the rules in the same format are substituted directly. For, negative and activation premises, substituted rules are in format (23) and, therefore, negative premises are substituted by negative premises, witness premises—by witness premises.  $\square$

### 3.3.4 Offer BIP

The offer predicate can be used instead of the transition relation for the definition of priorities. This offer semantics has been extensively studied in [4], where we have shown, among others, that instead of separating interaction and priority models, the same composition operators can be equivalently defined by including *negative* and *activation port typings* into interactions. Thus priorities are incorporated into *extended* interaction models, which can be encoded as Boolean formulas. In this paper we use this equivalent definition, since the Boolean encoding facilitates some of the proofs of expressiveness relations below. Notice, however, that this choice is purely a matter of convenience, since priority models can be defined explicitly in a straightforward manner, based on the inhibiting relation as in Section 3.2.3.

For a set of ports  $P$ , we denote  $\dot{P} \stackrel{\text{def}}{=} \{\dot{p} \mid p \in P\}$  and  $\bar{P} \stackrel{\text{def}}{=} \{\bar{p} \mid p \in P\}$ . We call the elements of  $P$ ,  $\dot{P}$  and  $\bar{P}$  *activation*, *firing* and *negative port typings* respectively.

**Definition 27** For a given set of ports  $P$ , an *extended interaction* is a subset  $a \subseteq P \cup \dot{P} \cup \bar{P}$ . An *extended interaction model* is a set of extended interactions  $\gamma \subseteq 2^{P \cup \dot{P} \cup \bar{P}}$ .

For a given extended interaction  $a$ , we define the following sets of ports:

- $\mathbf{act}(a) \stackrel{\text{def}}{=} a \cap P$ , the *activation support* of  $a$ ,
- $\mathbf{fire}(a) \stackrel{\text{def}}{=} \{p \in P \mid \dot{p} \in a\}$ , the *firing support* of  $a$ ,
- $\mathbf{neg}(a) \stackrel{\text{def}}{=} \{p \in P \mid \bar{p} \in a\}$ , the *negative support* of  $a$ .

**Definition 28** An *offer BIP* glue operator is defined by  $((P_i)_{i=1}^n, \gamma)$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and, denoting  $P \stackrel{\text{def}}{=} \bigcup_{i=1}^n P_i$ ,  $\gamma \subseteq 2^{P \cup \dot{P} \cup \bar{P}}$  is an extended interaction model.

The semantics is defined for a set of components  $C_i$  such that  $\sigma(C_i) = (Q_i, P_i, \rightarrow, \uparrow)^6$ , with  $i \in [1, n]$  by putting  $\sigma(\gamma(C_1, \dots, C_n)) \stackrel{\text{def}}{=} (Q, P, \rightarrow_\gamma, \uparrow_\gamma)$  with  $Q = \prod_{i=1}^n Q_i$ , the offer predicate  $\uparrow_\gamma$  defined by  $q_1 \dots q_n \uparrow_\gamma p \stackrel{\text{def}}{\iff} \exists i \in [1, n] : q_i \uparrow p$  and the transition relation  $\rightarrow_\gamma$  inductively defined by the rule

$$\frac{a \in \gamma \quad \left\{ q_i \xrightarrow{\mathbf{fire}(a) \cap P_i} q'_i \mid i \in I \right\} \quad \left\{ q_i = q'_i \mid i \notin I \right\} \quad \left\{ q_i \uparrow (\mathbf{act}(a) \cap P_i) \mid i \in [1, n] \right\} \quad \left\{ q_i \nmid p \mid i \in [1, n], p \in \mathbf{neg}(a) \cap P_i \right\}}{q_1 \dots q_n \xrightarrow{\mathbf{fire}(a)}_\gamma q'_1 \dots q'_n}, \quad (24)$$

where  $I = \{i \in [1, n] \mid \mathbf{fire}(a) \cap P_i \neq \emptyset\}$ .

We will denote by OBIP the framework, which has the same semantic domain and atomic components as FNASOS, but uses the set of offer BIP glue operators with the semantic mapping defined by (24).

<sup>6</sup> As in Note 1, we omit the indices on  $\uparrow$ , whenever they are clear from the context.

### 3.3.5 Activation BIP

Finally, for the sake of comparison, we consider another framework, which mixes the classical and offer semantics by relying on the usual transition relation to define the semantics of priority models, but using the offer predicate for that of non-firing positive premises.

**Definition 29** For given sets of ports  $P$ , an *interaction with activation* is a subset  $a \subseteq P \cup \dot{P}$ . An *interaction model with activation* is a set  $\gamma \subseteq 2^{P \cup \dot{P}}$  of interactions with activation.

**Definition 30** An *activation BIP* glue operator is defined by  $((P_i)_{i=1}^n, \gamma, \pi)$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports,  $\gamma \subseteq 2^{P \cup \dot{P}}$  is an interaction model with activation and  $\pi \subseteq 2^{2^{P \cup \dot{P}} \setminus \{\emptyset\}} \setminus \{\emptyset\}$  is a priority model.

The semantics of the application of an operator  $((P_i)_{i=1}^n, \gamma, \pi)$  is defined for any set of components  $C_1, \dots, C_n$  such that  $\sigma(C_i) = (Q_i, P_i, \rightarrow_i, \uparrow_i)$ , for  $i \in [1, n]$ , by putting  $\sigma(\pi\gamma\langle C_1, \dots, C_n \rangle) \stackrel{\text{def}}{=} (Q, P, \rightarrow_{\pi\gamma}, \uparrow_{\pi\gamma})$ , with  $Q = \prod_{i=1}^n Q_i$  and the minimal transition relation  $\rightarrow_{\pi\gamma}$  satisfying the rule

$$\left\{ \begin{array}{l} \{q_i \xrightarrow{\text{fire}(a) \cap P_i} q'_i \mid i \in I\} \\ \{q_i = q'_i \mid i \notin I\} \\ \left\{ q_{j(\alpha)} \xrightarrow{b(\alpha) \cap P_{j(\alpha)}} \mid \alpha \in \pi(a) \right\} \\ \{q_k \uparrow (a \cap P_k) \mid k \in K\} \\ \hline q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n \end{array} \middle| \begin{array}{l} a \in \gamma, \\ b : \pi(a) \Rightarrow \bigcup \pi(a) : (\alpha).b \in \alpha, \\ j : \pi(a) \Rightarrow [1, n] : (\alpha).b(\alpha) \cap P_j \neq \emptyset \end{array} \right\}, \quad (25)$$

where  $I = \{i \in [1, n] \mid a \cap \dot{P}_i \neq \emptyset\}$  and  $K = \{k \in [1, n] \mid a \cap P_k \neq \emptyset\}$ . The offer predicate  $\uparrow_{\pi\gamma}$  is defined as above by  $q_1 \dots q_n \uparrow_{\pi\gamma} p \stackrel{\text{def}}{\iff} \exists i \in [1, n] : q_i \uparrow_i p$ .

We will denote by ABIP the framework, which has the same semantic domain and atomic components as FNASOS, but uses the set of activation BIP glue operators with the semantic mapping defined by (25).

## 4 Proofs of the expressiveness relations

In this section we prove the relations depicted in Figure 1. First, in Section 4.1, we show in-depth comparison of classical BIP and simple BIP-like SOS explaining the limitations of CBIP and how one could circumvent them. Then, in Section 4.2, we prove the rest of relations between the different frameworks.

*Note 5* Since the first element of all composition operators is the disjoint sets of ports  $(P_i)_{i=1}^n$ , we will omit mentioning it explicitly in most of the following proofs. Indeed, if we want to apply two composition operators to the same sets of behaviours, their disjoint sets of ports have to be equal. Instead, we will speak of *composition operators over  $P$* , where  $P = \bigcup_{i=1}^n P_i$ , with  $(P_i)_{i=1}^n$  being mostly implicit.

By Lemma 3, all the frameworks considered in the paper are compositional. Furthermore, they all share the same set of atomic components and—the extension of behaviours with the offer predicate being canonical—the same semantic domain. Therefore, to prove strong/weak full expressiveness relations we will rely on Proposition 3. Thus we only have to exhibit operators that, given pairs of sets of components where corresponding pairs of components have the *same* behaviour, produce components with the same behaviour. In this context, the sets of states and ports, as well as offer predicates of the composed behaviours coincide trivially. Thus, in the proofs we will only check the equality of transition relations. The following lemma shows that it is not necessary to consider target states of transitions and it is sufficient to compare labels of outgoing transitions for each state of composed systems.

**Lemma 5** *Let  $(C_i)_{i=1}^n$  be a set of components, with  $\sigma(C_i) = (Q_i, P_i, \rightarrow, \uparrow)$ , and let  $P = \bigcup_{i=1}^n P_i$ . Let  $o_1$  and  $o_2$  be composition operators over  $P$ . Then, the following statement holds for behaviours of composed systems  $\sigma(o_1\langle C_1, \dots, C_n \rangle) = (Q, P, \rightarrow_{o_1}, \uparrow)$  and  $\sigma(o_2\langle C_1, \dots, C_n \rangle) = (Q, P, \rightarrow_{o_2}, \uparrow)$ : for any state  $q \in Q$  and for any transition label  $a \in 2^P$ , if  $q \xrightarrow{a}_{o_1} \Leftrightarrow q \xrightarrow{a}_{o_2}$  then  $\{q' \mid (q, a, q') \in \rightarrow_{o_1}\} = \{q' \mid (q, a, q') \in \rightarrow_{o_2}\}$ .*

*Proof.* If  $q \not\xrightarrow{a}_{o_1}$  then  $q \not\xrightarrow{a}_{o_2}$  and both sets are empty.

Suppose now that  $(q, a, q') \in \rightarrow_{o_1}$ , for some  $q' \in Q$ . Then  $q \xrightarrow{a}_{o_1} q'$  and, by the assumption of the lemma  $q \xrightarrow{a}_{o_2}$ . Furthermore, there exists a corresponding interaction  $a$ , interaction with activation or extended interaction  $b$  with  $\mathbf{fire}(b) = a$ , or a rule  $r$  with the conclusion  $q \xrightarrow{a}_{o_1} q'$ . Let  $q = (q_i)_{i=1}^n$  and  $q' = (q'_i)_{i=1}^n$  and notice that, for all BIP glues and all considered SOS rule formats,  $(q, a, q') \in \rightarrow_{o_1}$  if, for all  $i \in [1, n]$ ,  $q_i \xrightarrow{a \cap P_i} q'_i$ , if  $a \cap P_i \neq \emptyset$ , and  $q_i = q'_i$  otherwise. Since  $q \xrightarrow{a}_{o_2}$ , there exists an enabled rule in the semantics of  $o_2$  with the conclusion  $a$ . Since the enabledness of the rule does not depend on the target states of the firing premises, we can consider the same firing premises as for  $o_1$ , i.e.  $q_i \xrightarrow{a \cap P_i} q'_i$ , if  $a \cap P_i \neq \emptyset$ . Hence,  $q \xrightarrow{a}_{o_2} q'$  and, therefore,  $\{q' \mid (q, a, q') \in \rightarrow_{o_1}\} = \{q' \mid (q, a, q') \in \rightarrow_{o_2}\}$ .  $\square$

#### 4.1 Expressiveness of the classical BIP

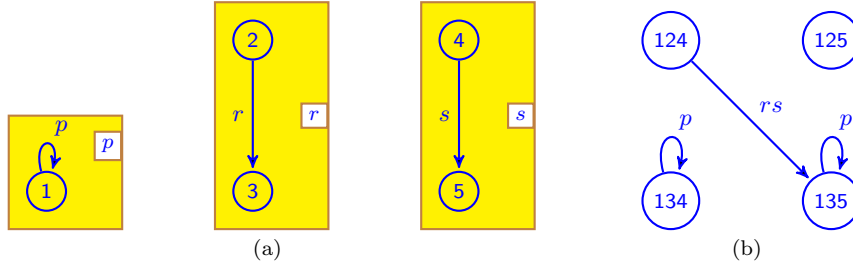
In [4, Example 2.9 and its discussion] we have shown an AcBSOS operator<sup>7</sup> that cannot be expressed as a CBIP operator. In the following example we provide a simplified version of the one in [4] and an intuitive justification.

*Example 8* Consider the BSOS operator defined by the following two rules:

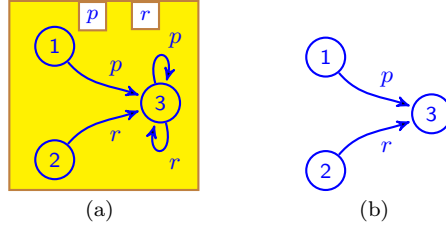
$$\frac{q_1 \xrightarrow{p} q'_1 \quad q_2 \xrightarrow{r} q'_2}{q_1 q_2 q_3 \xrightarrow{p} q'_1 q_2 q_3}, \quad \frac{q_2 \xrightarrow{r} q'_2 \quad q_3 \xrightarrow{s} q'_3}{q_1 q_2 q_3 \xrightarrow{rs} q_1 q'_2 q'_3} \quad (26)$$

with the port partition  $P = (\{p\}, \{r\}, \{s\})$ . Figure 4(b) shows the behaviour resulting from the application of this operator to the components in Figure 4(a). If

<sup>7</sup> We have only introduced the term “AcBSOS” in the present paper. In [4], we speak of composition operators defined by sets of SOS rules.



**Fig. 4** Component and behaviour for Example 8



**Fig. 5** Component and behaviour for Example 9

this operator were to be realised by a single CBIP operator, then necessarily its interaction model would be  $\gamma = \{p, rs\}$ . Since  $rs$  is only enabled in the state 124, there is no way to define the priority model so as to suppress  $p$  in the states 124 and 125, but not in 134 and 135.

Since operator (26) is simple and acyclic, we conclude that CBIP does not have strong full expressiveness w.r.t. SiBSOS and AcBSOS.

The example below shows that CBIP does not have even weak full expressiveness w.r.t. SiBSOS, whereas the subsequent proposition shows that it does have it w.r.t. AcBSOS.

*Example 9* Consider a composition operator defined by the following two rules:

$$\frac{q_1 \xrightarrow{p} q_1' \quad q_1 \xrightarrow{r} q_1'}{q_1 \xrightarrow{p} q_1'}, \quad \frac{q_1 \xrightarrow{r} q_1' \quad q_1 \xrightarrow{p} q_1'}{q_1 \xrightarrow{r} q_1'}, \quad (27)$$

applied to the component shown in Figure 5(a). The behaviour of the composed component is shown in Figure 5(b). Assume that there exists a hierarchy of BIP glues, such that applying them to the same component results in an equivalent composed component. States 1 and 2 of the composed component behaviour have outgoing transitions  $p$  and  $r$ , respectively, thus all interaction models in the glues have to contain both interactions  $p$  and  $r$ . State 3 is a deadlock. Interaction models do not forbid any transition from this state and priority models cannot introduce deadlock by Lemma 4. This contradicts the assumption and, consequently, the set of rules (27) is not expressible in CBIP.

The two fundamental reasons for this lack of expressiveness are related to the definition of the priority model:

- the information used by the priority model refers only to interactions authorised by the underlying interaction model—all the information about transitions enabled in sub-components is lost [4];
- the priority model  $\pi$  must be a strict partial order.

As we explain below, among these two reasons, the first one is easily addressed to achieve weak, rather than strong, full expressiveness, whereas the second one presents the main difficulty.

*What can be done without changing the BIP glue?* As in Section 3.2.3, consider an  $n$ -ary BIP-like SOS operator  $o = ((P_i)_{i=1}^n, R)$  with  $R$  a non-redundant set of rules

$$\frac{\left\{ q_i \xrightarrow{a^r \cap P_i} q'_i \mid i \in I^r \right\} \quad \left\{ q_i = q'_i \mid i \notin I^r \right\} \quad \left\{ q_j \xrightarrow{b_j^r} (j, b_j^r) \in H^r \right\}}{q_1 \dots q_n \xrightarrow{a^r} q'_1 \dots q'_n}, \quad (28)$$

for each  $r \in R$ , where, as before,  $I^r = \{i \in [1, n] \mid a^r \cap P_i \neq \emptyset\}$ ,  $H^r \subseteq [1, n] \times 2^P$  and, for each  $(j, b_j^r) \in H^r$ , holds  $b_j^r \in 2^{P_j}$ . For an interaction  $a \in \{a^r \mid r \in R\}$ , denote  $R_a \stackrel{\text{def}}{=} \{r \in R \mid a^r = a \wedge H^r \neq \emptyset\}$  the set of rules with negative premises that have the conclusion labelled by  $a$ . Let  $\pi$  be the inhibiting relation of  $o$  (Definition 16).

*Note 6* As for the transition relations and offer predicates, we will henceforth omit the indices on the semantic mappings  $\sigma$ , since they will be clear from the context. For instance, in the proof of the proposition below,  $\sigma$  will refer to both the semantic mapping of CBIP and that of AcBSOS.

**Proposition 6** *If  $\pi$  is simple and acyclic, then the operator  $o$  can be realised by a hierarchical composition of BIP glue operators.*

*Proof.* Since  $\pi$  is acyclic, we can associate a depth  $d(a)$  to each interaction  $a$  involved in  $\pi$  as the length of the longest path leading to  $a$  in the directed acyclic graph defined by  $\pi$ . Denote  $d \stackrel{\text{def}}{=} \max_a d(a)$ . Furthermore, for  $i \in [1, d]$ , denote  $\pi_i \stackrel{\text{def}}{=} \{(a, b) \in \pi \mid d(a) = i - 1\}$ .

Clearly all  $\pi_i$  are strict partial orders. Furthermore  $\pi_i \subseteq \pi \subseteq \gamma_1 \times \gamma_1$ , for all  $i \in [1, d]$  and

$$\gamma_1 = \gamma_2 \cup \{b \mid a \in \gamma_2, (a, b) \in \pi\}, \text{ where } \gamma_2 = \{a^r \mid r \in R\}.$$

Hence, for all  $i \in [1, d]$ ,  $(\gamma_1, \pi_i)$  is a BIP glue operator.

The operator  $o$  is equivalent to the composition  $(\gamma_2, \emptyset) \circ (\gamma_1, \pi_d) \circ \dots \circ (\gamma_1, \pi_1)$ . We show that for any set of components  $C_i$  with  $\sigma(C_i) = (Q_i, P_i, \rightarrow)$ , for  $i \in [1, n]$ , holds  $\sigma(\gamma_2 \langle \pi_d \gamma_1 \langle \dots \pi_1 \gamma_1 \langle C_1, \dots, C_n \rangle \dots \rangle \rangle) = \sigma(o \langle C_1, \dots, C_n \rangle)$ . We denote

$$B_o = \sigma(o \langle C_1, \dots, C_n \rangle), \quad B_{\pi\gamma} = \sigma(\gamma_2 \langle \pi_d \gamma_1 \langle \dots \pi_1 \gamma_1 \langle C_1, \dots, C_n \rangle \dots \rangle \rangle).$$

The sets of states and ports of these behaviours are the same, thus we need to check only that their transition relations coincide.

Let  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_o$ . This means that, among the rules defining  $o$ , i.e. for some  $r \in R$ , there is a rule

$$\frac{\left\{ q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I^r \right\} \quad \left\{ q_i = q'_i \mid i \notin I^r \right\} \quad \left\{ q_j \xrightarrow{b_j^r} (j, b_j^r) \in H^r \right\}}{q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n}, \quad (29)$$

such that  $q_i \xrightarrow{a \cap P_i}$ , for all  $i \in I$ , and  $q_j \xrightarrow{b_j^r}$  for all  $(j, b_j^r) \in H^r$ . By construction both  $\gamma_1$  and  $\gamma_2$  contain  $a$ . Hence,  $a$  is enabled in the state  $q_1 \dots q_n$  of  $\sigma(\gamma_1 \langle C_1, \dots, C_n \rangle)$  and in the same state of  $B_{\pi\gamma}$ , provided that it is not disabled by any of priorities  $\pi_1, \dots, \pi_d$ . Thus, we have to show that no interaction available from this state has higher priority. By construction, priority rules that contain  $a$  in the left-hand side can appear only in  $\pi_{d(a)-1}$ , thus other priority models cannot block  $a$ . Priority rules of the form  $a < b$  have  $b = \bigcup_{s \in R_a} b_{j(s)}^s$ , for some  $j : R_a \Rightarrow [1, n] : (s). (j, b_j^s) \in H^s$ . Since all the premises of (29) are satisfied in  $q_1 \dots q_n$ , interaction  $b_{j(r)}^r$  is disabled. Hence,  $b$  is also disabled. Thus  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_{\pi\gamma}$ .

Let  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_{\pi\gamma}$ . This means that both  $\gamma_1$  and  $\gamma_2$  contain the interaction  $a$ . Therefore, by the construction of  $\gamma_2$ , there is at least one rule

$$\frac{\left\{ q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I \right\} \quad \left\{ q_i = q'_i \mid i \notin I \right\} \quad \left\{ q_j \xrightarrow{b_j} (j, b_j) \in H \right\}}{q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n}, \quad (30)$$

with conclusion labelled by  $a$  among the rules defining  $o$ . Furthermore, the priority model  $\pi_{d(a)-1}$  contains priorities of the form  $a < b$ , with  $b = \bigcup_{s \in R_a} b_{j(s)}^s$ , for all  $j : R_a \Rightarrow [1, n] : (s). (j, b_j^s) \in H^s$ . Notice that a priority rule  $b < c$  such that  $a < b$  cannot appear in priorities  $\pi_1, \dots, \pi_{d(a)-1}$  since  $d(b) \geq d(a)+1$ . Assume that none of rules defining  $o$ , with the conclusion labelled by  $a$ , applies in  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$ . This necessarily means that each of these rules has a negative premise that is not satisfied. Let  $b = \bigcup_{s \in R_a} b_{j(s)}^s$  with  $b_{j(s)}^s$ , for all  $s \in R_a$ , being the labels of dissatisfied premises. Then  $b$  is an enabled interaction in  $\sigma(\gamma_1 \langle C_1, \dots, C_n \rangle)$  such that  $a < b$  and  $b$  cannot be blocked by priorities  $\pi_1, \dots, \pi_{d(a)-1}$ . Consequently,  $b$  is enabled in  $\sigma(\pi_{d(a)-1} \gamma_1 \langle \dots \pi_1 \gamma_1 \langle C_1, \dots, C_n \rangle \dots \rangle)$  and blocks  $a$ , which contradicts the assumption  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_{\pi\gamma}$ . Hence, there is at least one rule of the form (30) in the definition of  $o$  with all premises satisfied in  $q_1 \dots q_n$  and, therefore,  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_o$ .  $\square$

Thus, CBIP has weak full expressiveness w.r.t. AcBSOS.

*What can be achieved by relaxing constraints on priority models?* Instead of imposing additional constraints on the set of BIP-like SOS operators we can relax the priority model of BIP.

**Definition 31** A *semi-relaxed BIP operator* is a triple  $((P_i)_{i=1}^n, \gamma, \pi)$ , where  $(P_i)_{i=1}^n$  are disjoint sets of ports and, denoting  $P = \bigcup_{i=1}^n P_i$ ,  $\gamma$  is an interaction model on  $P$  and  $\pi$  is a relaxed priority model on  $\gamma$ ,  $\pi \subseteq \gamma \times \gamma$ .

Notice that the priority model is not required to be a partial order but, contrary to the relaxed BIP (Definition 21), it can involve only interactions from the interaction model. Its semantics is defined exactly as that of classical priority models, by (8).

Given a simple BIP-like SOS operator  $o = ((P_i)_{i=1}^n, R)$ , we consider its inhibiting relation  $\pi$  and the interaction models  $\gamma_1, \gamma_2$  as in the proof of Proposition 6. Since  $\pi \subseteq \gamma_1 \times \gamma_1$ , the operator  $(\gamma_1, \pi)$  is a semi-relaxed BIP operator. The operator  $o$  is then equivalent to the composition  $(\gamma_2, \emptyset) \circ (\gamma_1, \pi)$ , where  $\pi$  is considered as a relaxed priority model.

**Proposition 7** *For any set of components  $(C_i)_{i=1}^n$  with  $\sigma(C_i) = (Q_i, P_i, \rightarrow)$ , for  $i \in [1, n]$ , holds*

$$\sigma(\gamma_2 \langle \pi \gamma_1 \langle C_1, \dots, C_n \rangle \rangle) = \sigma(o \langle C_1, \dots, C_n \rangle).$$

*Proof.* Denote  $B_o = \sigma(o \langle C_1, \dots, C_n \rangle)$  and  $B_{\pi\gamma} = \sigma(\gamma_2 \langle \pi \gamma_1 \langle C_1, \dots, C_n \rangle \rangle)$ . The sets of states and ports of these behaviours are the same. Thus, we need to check only that their transitions coincide.

Let  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_o$ . This means that, among the rules defining  $o$ , i.e. for some  $r \in R$ , there is a rule

$$\frac{\left\{ q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I^r \right\} \quad \left\{ q_i = q'_i \mid i \notin I^r \right\} \quad \left\{ q_j \xrightarrow{b_j^r} (j, b_j^r) \in H^r \right\}}{q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n}, \quad (31)$$

with all its premises satisfied for the respective behaviours  $\sigma(C_i)$ . By construction both  $\gamma_1$  and  $\gamma_2$  contain  $a$ . Hence,  $a$  is enabled in the state  $q_1 \dots q_n$  of  $\sigma(\gamma_1 \langle C_1, \dots, C_n \rangle)$  and in the same state of  $\sigma(\gamma_2 \langle \pi \gamma_1 \langle C_1, \dots, C_n \rangle \rangle)$ , provided that it is not disabled by the priority  $\pi$ . Thus, we have to show that no interaction available from this state has higher priority.

Priority rules in  $\pi$  that contain  $a$  are of the form  $a < b$ , with  $b = \bigcup_{s \in R_a} b_{j(s)}^s$ , for some  $j : R_a \rightrightarrows [1, n] : (s). (j, b_j^s) \in H^s$ . Since all the premises of (31) are satisfied in  $q_1 \dots q_n$ , interaction  $b_{j(r)}^r$  is disabled. Hence,  $b$  is also disabled. Thus  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_{\pi\gamma}$ .

Let  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_{\pi\gamma}$ . This means that both  $\gamma_1$  and  $\gamma_2$  contain the interaction  $a$ . Therefore, by the construction of  $\gamma_2$ , there is at least one rule

$$\frac{\left\{ q_i \xrightarrow{a \cap P_i} q'_i \mid i \in I \right\} \quad \left\{ q_i = q'_i \mid i \notin I \right\} \quad \left\{ q_j \xrightarrow{b_j} (j, b_j) \in H \right\}}{q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n}, \quad (32)$$

among the rules defining  $o$ . Furthermore, the priority model  $\pi$  has to contain priorities of the form  $a < b$ , with  $b = \bigcup_{s \in R_a} b_{j(s)}^s$ , for some  $j : R_a \rightrightarrows [1, n] : (s). (j, b_j^s) \in H^s$ . Assume now that none of the rules defining  $o$ , with the conclusion labelled by  $a$ , applies in  $q_1 \dots q_n$ . Since  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_{\pi\gamma}$ , this necessarily means that each of these rules has a negative premise that is not satisfied. Let  $b = \bigcup_{s \in R_a} b_{j(s)}^s$  with  $b_{j(s)}^s$ , for all  $s \in R_a$ , being the labels of dissatisfied premises. Then  $b$  is an enabled interaction such that  $a < b$ , which contradicts the assumption  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_{\pi\gamma}$ . Hence, there is at least one rule of the form (32) in the definition of  $o$  with all premises satisfied in  $q_1 \dots q_n$  and, therefore,  $q_1 \dots q_n \xrightarrow{a} q'_1 \dots q'_n$  in  $B_o$ .  $\square$

Thus, we conclude that semi-relaxed BIP (same as RBIP, but with semi-relaxed glue operators instead of relaxed ones) has weak full expressiveness w.r.t. SiBSOS.

Notice that the relaxed priority model defined over the interaction model does not allow recovering strong full expressiveness. For instance, consider the operator defined by the single rule

$$\frac{q_1 \xrightarrow{p} q_1' \quad q_1 \xrightarrow{r} q_1'}{q_1 \xrightarrow{p} q_1'}, \quad (33)$$

applied to a component with the behaviour in Figure 5. The composed component behaviour has a single transition  $1 \xrightarrow{p} 3$ . The interaction model of BIP cannot contain  $r$ , as it is not possible to exclude transition  $2 \xrightarrow{r} 3$  with a priority model. The transition  $3 \xrightarrow{p} 3$  has to be excluded by the priority model, however it cannot use  $r$  in the priority model.

Further relaxation of the definition of BIP operators by removing the restriction  $\pi \subseteq \gamma \times \gamma$  results exactly in the relaxed BIP (Definition 21). With this relaxation we obtain strong full expressiveness w.r.t. SiBSOS, since any simple BIP-like SOS operator  $o$  is then clearly equivalent to  $(\gamma_2, \pi)$  with  $\gamma_2$  containing interactions that label conclusions of the rules defining  $o$  (as in the proof of Proposition 6), while  $\pi$  is the inhibiting relation of  $o$ . Thus, RBIP has strong full expressiveness w.r.t. SiBSOS.

Consider another relaxation of the definition of BIP glue operators, by taking operators  $((P_i)_{i=1}^n, \gamma, \pi)$ , with  $P = \bigcup_{i=1}^n P_i$ , such that the priority model  $\pi \subseteq 2^P \times (2^P \setminus \emptyset)$  is a strict partial order, without requiring that it refers only to interactions (i.e. we do not impose  $\pi \subseteq \gamma \times \gamma$ ). This relaxation does not recover even weak full expressiveness w.r.t. simple BIP-like SOS operators. Indeed, Example 9 is still not expressible.

#### 4.2 Comparison of the modifications of BIP and SOS formats

In this section, we provide the proofs for the expressiveness comparison between various BIP glues and SOS rules formats shown in Figure 1.

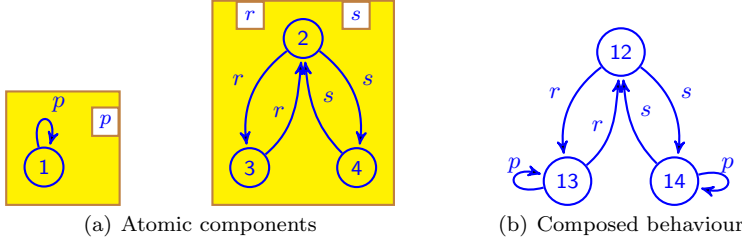
**CBIP  $\rightarrow$  AcBSOS and CBIP  $\Rightarrow$  SiBSOS:** As shown in Section 4.1, CBIP has weak full expressiveness w.r.t. AcBSOS and does not have even weak full expressiveness w.r.t. SiBSOS. Example 8 shows that CBIP does not have strong full expressiveness w.r.t. AcBSOS.

Since the semantic mapping of CBIP is defined through acyclic BIP-like SOS (see Note 3), both AcBSOS and SiBSOS have strong full expressiveness w.r.t. CBIP.

**RBIP  $\Leftrightarrow$  SiBSOS:** As observed in Section 4.1, RBIP has strong full expressiveness w.r.t. SiBSOS. On the other hand, the semantic mapping of RBIP is defined through sets of rules in the BIP-like SOS format with simple inhibiting relations. Thus SiBSOS also has strong full expressiveness w.r.t. RBIP.

**CBIP  $\Rightarrow$  RBIP:** Follows from CBIP  $\Rightarrow$  SiBSOS and RBIP  $\Leftrightarrow$  SiBSOS by Corollary 1.





**Fig. 6** Component and behaviour for Example 10

**RBIP  $\Rightarrow$  XBIP:** XBIP is an extension of RBIP, thus it has strong full expressiveness w.r.t. RBIP. The following example shows that RBIP does not have weak full expressiveness w.r.t. XBIP.

*Example 10* Consider an XBIP operator  $(\{\{p\}, \{r, s\}\}, \gamma = \{p, r, s\}, \pi = \{(p, \{r, s\})\})$  applied to components shown in Figure 6(a). The behaviour of the composed component is shown in Figure 6(b). Assume there exists a hierarchy of RBIP operators such that applying them to the same components results in an equivalent composed one. Clearly, all interaction models have to involve  $p$ , thus transition  $p$  from the state 12 must be blocked by a relaxed priority model. If this priority model contains  $p < r$  or  $p < pr$ , then the transition  $p$  would also be blocked in the state 13. Similarly, for  $p < s$  or  $p < ps$  and the state 14. Priority rules  $p < rs$  and  $p < prs$  cannot be applied in any state, since they require that transition  $rs$  be available (see (20) and notice that  $r$  and  $s$  belong to the interface of the same component). Thus, no priority rule can block  $p$  only in the state 12.

**XBIP  $\Leftrightarrow$  BSOS:** The semantic mapping of XBIP is defined in terms of the BIP-like SOS format, implying that BSOS has strong full expressiveness w.r.t. XBIP. On the other hand, given any BIP-like SOS composition operator  $o = ((P_i)_{i=1}^n, \mathcal{R})$ , the XBIP operator  $((P_i)_{i=1}^n, \gamma, \pi)$ , where  $\gamma$  is the set of interactions labelling the conclusions of the rules in  $\mathcal{R}$  and  $\pi$  is the inhibiting relation, is clearly equivalent to  $o$ . Hence, XBIP also has strong full expressiveness w.r.t. BSOS.

**AcBSOS  $\Rightarrow$  SiBSOS:** Since AcBSOS is a restriction of SiBSOS, the latter has strong full expressiveness w.r.t. the former. On the other hand, AcBSOS does not have weak full expressiveness w.r.t. SiBSOS. Indeed, that would imply that so does CBIP, which is not the case, since we have already established  $CBIP \Rightarrow$  SiBSOS.

**SiBSOS  $\Rightarrow$  BSOS:** Follows from  $SiBSOS \Leftrightarrow RBIP$ ,  $RBIP \Rightarrow XBIP$  and  $XBIP \Leftrightarrow BSOS$  by Corollary 1.

**XBIP  $\Rightarrow$  ABIP:** Clearly, ABIP is an extension of XBIP. Thus, it has strong full expressiveness w.r.t. XBIP. The following example shows that BSOS does not have weak full expressiveness w.r.t. ABIP. By Lemma 1, since  $XBIP \Leftrightarrow BSOS$ , this implies that XBIP does not have weak full expressiveness w.r.t. ABIP either.

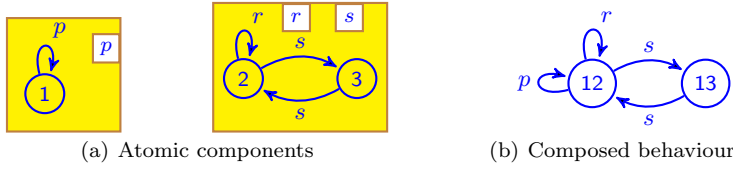


Fig. 7 Components and behaviour for Example 11

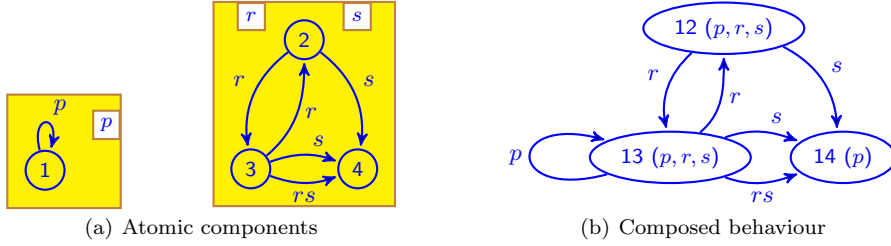


Fig. 8 Components and behaviour for Example 12

*Example 11* Consider two components in Figure 7(a) and the activation BIP glue operator  $((\{p\}, \{r, s\}), \{\dot{p}r, \dot{r}, \dot{s}\}, \emptyset)$ . The semantics of this operator is given by the rules

$$\frac{q_1 \xrightarrow{p} q_1' \quad q_2 \uparrow r}{q_1 q_2 \xrightarrow{p} q_1' q_2}, \quad \frac{q_2 \xrightarrow{r} q_2' \quad q_1 \uparrow}{q_1 q_2 \xrightarrow{r} q_1 q_2'}, \quad \frac{q_2 \xrightarrow{s} q_2' \quad q_1 \uparrow}{q_1 q_2 \xrightarrow{s} q_1 q_2'}. \quad (34)$$

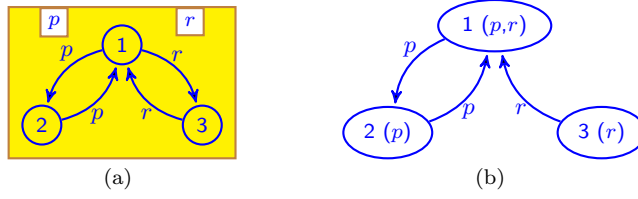
The composed component behaviour (shown in Figure 7(b) without the offer predicate) has a transition  $p$  only in the state 12. Suppose that there exists a hierarchy of BIP-like SOS operators with the equivalent semantics. All of them must contain rules allowing transitions  $r$  and  $s$ . However it is not possible to allow transition  $p$  from state 12 but not from state 13, since the only negative premise that can be used to block it is  $q_2 \not\xrightarrow{s}$  and it would block  $p$  in both states.

**ABIP  $\leftrightarrow$  WBSOS:** Activation BIP does not have weak full expressiveness w.r.t. WBSOS as shown in the following example.

*Example 12* Consider two components shown in Figure 8(a) and the WBSOS operator defined by the following set of rules:

$$\frac{q_1 \xrightarrow{p} q_1' \quad q_2 \xrightarrow{rs}}{q_1 q_2 \xrightarrow{p} q_1' q_2}, \quad \frac{q_2 \xrightarrow{r} q_2' \quad q_1 \uparrow}{q_1 q_2 \xrightarrow{r} q_1 q_2'}, \quad \frac{q_2 \xrightarrow{s} q_2' \quad q_1 \uparrow}{q_1 q_2 \xrightarrow{s} q_1 q_2'}, \quad \frac{q_2 \xrightarrow{rs} q_2' \quad q_1 \uparrow}{q_1 q_2 \xrightarrow{rs} q_1 q_2'}.$$

Consider the states 12 and 13 of the composed component behaviour (shown in Figure 8(b) with ports offered in each state shown in parentheses): the former does not have an outgoing transition  $p$  while the latter does. In ABIP any interaction with the firing support  $p$  would allow the transition  $p$  from both states, since the two states are indistinguishable in terms of the offer predicate. Thus the transition  $p$  from the state 12 has to be blocked by a relaxed priority model. However, any priority rule blocking this transition in the state 12 also blocks it in the state



**Fig. 9** An atomic component (a) and the behaviour after application of the first glue operator (b) in Examples 13, 15 and 16 (ports offered in each state are shown in parentheses)

13 since the set of enabled transitions in the state 12 is a subset of the set of transitions enabled in the state 13. Thus, this system is inexpressible in ABIP.

The converse is shown by another example.

*Example 13* Consider the atomic component  $C$  shown in Figure 9(a) and an activation BIP glue operator  $o_1 = (\{p, r\}, \gamma, \pi)$ , with  $\gamma = \{\dot{p}, \dot{r}\}$  and  $\pi = \{r < \{p\}\}$ . The behaviour of  $o_1\langle C \rangle$  is shown in Figure 9(b).

Consider an activation BIP operator  $o_2 = (\{p, r\}, \gamma_2, \pi_2)$ , with  $\gamma_2 = \{\dot{p}r\}$  and  $\pi = \emptyset$  applied to the composed component  $o_1\langle C \rangle$ . Both ports  $p$  and  $r$  are offered in the state 1 of  $\sigma(o_1\langle C \rangle)$ , thus the interaction  $\dot{p}r$  is enabled in this state despite the fact that the state does not have any outgoing transition involving  $r$ . In the state 2,  $r$  is not offered and the interaction is not enabled.

Consider now the (witness) BIP-like SOS operator  $o'_1$  defined by the rules

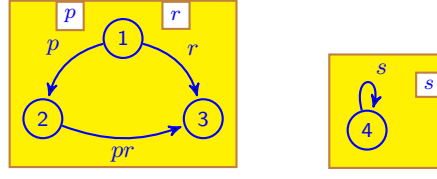
$$\frac{q \xrightarrow{p} q'}{q \xrightarrow{o'_1} q'}, \quad \frac{q \xrightarrow{r} q' \quad q \xrightarrow{p} q'}{q \xrightarrow{o'_1} q'}.$$

Clearly,  $\sigma(o'_1\langle C \rangle) = \sigma(o_1\langle C \rangle)$ . However, it is not possible to construct a witness BIP-like SOS operator that applied to  $o'_1\langle C \rangle$  would result in a component with behaviour equivalent to  $\sigma(o_2\langle o_1\langle C \rangle \rangle)$ . Indeed, it is not possible to distinguish states 1 and 2, since they have the same sets of outgoing transitions in  $\sigma(o_1\langle C \rangle)$ . Thus, WBSOS does not have strong full expressiveness w.r.t. AcBSOS and, since WBSOS possesses uniform flattening, it does not have weak full expressiveness either.

**BSOS  $\Rightarrow$  WBSOS:** Witness BSOS is an extension of BSOS, thus, trivially, it has strong full expressiveness w.r.t. BSOS. Example 12 presents a system with a WBSOS operator inexpressible in ABIP and, consequently, in BSOS.

**OBIP  $\Leftrightarrow$  FNASOS:** It is clear by comparing (22) and (24) that any offer BIP operator can be encoded in FNASOS. The inverse also holds: for any SOS rule in the FNASOS format (24), we can provide a corresponding extended interaction, where firing, activation and negation supports are obtained from firing, activation and negative premises respectively. Since FNASOS possesses uniform flattening we can conclude the same for OBIP.

**Corollary 3** *OBIP possesses uniform flattening.*



**Fig. 10** Atomic components for Example 14

**CBIP  $\leftrightarrow$  OBIP:** The incomparability of CBIP and OBIP was shown in [4]. The same reasoning as in Example 9 proves that the offer BIP composition operator  $(\{p, r\}, \gamma)$  with  $\gamma = \{\dot{p}\bar{r}, \dot{r}\bar{p}\}$ , whereof the semantics is given by the rules

$$\frac{q_1 \xrightarrow{p} q_1' \quad q_1 \nrightarrow r}{q_1 \xrightarrow{p} q_1'}, \quad \frac{q_1 \xrightarrow{r} q_1' \quad q_1 \nrightarrow p}{q_1 \xrightarrow{r} q_1'},$$

is inexpressible in CBIP. The following example from [4] presents a system inexpressible in OBIP.

*Example 14* Consider the atomic components shown in Figure 10 and the classical BIP composition operator  $(\{\{p, r\}, \{s\}\}, \gamma, \pi)$  with the interaction model  $\gamma = \{p, r, pr, s\}$  and priority model  $\pi = \{s < pr\}$ . Since, classical priority semantics refers to the enabledness of an interaction, in the behaviour of the composed system the interaction  $s$  is available in the state 14, but not in the state 24. In the offer semantics, all three ports are offered in both states 14 and 24 of this system. Therefore, these two states are indistinguishable and  $s$  cannot be inhibited in only one of them by any hierarchical composition of OBIP operators.

**OBIP  $\leftrightarrow$  ABIP:** Example 14 shows that OBIP does not have weak full expressiveness w.r.t. CBIP. Since we have  $\text{CBIP} \Rightarrow \text{RBIP} \Rightarrow \text{XBIP} \Rightarrow \text{ABIP}$ , this means that OBIP does not have weak full expressiveness w.r.t. ABIP either. The following example shows that ABIP does not have weak full expressiveness w.r.t. OBIP, i.e. OBIP and ABIP are incomparable.

*Example 15* Consider an atomic component  $C$  shown in Figure 9(a). Both ports  $p$  and  $r$  are offered in the state 1, while only  $p$  is offered in the state 2. Consider the offer BIP glue operator  $o_1 = (\{p, r\}, \gamma)$ , with  $\gamma = \{\dot{p}, \dot{r}\bar{p}\}$ . The behaviour of the composed component is shown in Figure 9(b). Notice that the offer predicates of  $\sigma(C)$  and  $\sigma(o_1\langle C \rangle)$  coincide.

Consider the offer BIP glue operator  $o_2 = (\{p, r\}, \gamma_2)$ , with  $\gamma_2 = \{\dot{p}\bar{r}\}$  applied to the composed component  $o_1\langle C \rangle$ . Both ports  $p$  and  $r$  are offered in the state 1, thus the interaction  $\dot{p}\bar{r}$  is not enabled in this state despite the fact that the state does not have any outgoing transition involving  $r$ .

Consider now the activation BIP operator  $o_1' = (\{p, r\}, \gamma, \pi)$ , with  $\gamma = \{\dot{p}, \dot{r}\}$  and  $\pi = \{r < \{p\}\}$ . Clearly  $\sigma(o_1'\langle C \rangle) = \sigma(o_1\langle C \rangle)$ . However, there is no activation BIP glue operator or a hierarchy of operators that, applied to  $o_1'\langle C \rangle$  would produce a component equivalent to  $o_2\langle o_1\langle C \rangle \rangle$ . Indeed, this would require forbidding the transition  $p$  from the state 1 without forbidding it from the state 2, as is the case in  $\sigma(o_2\langle o_1\langle C \rangle \rangle)$ . Indeed, we have to keep  $\dot{p}$  in the interaction model, since  $2 \xrightarrow{p} 1$ , but there are no transitions from state 1 that could be used to inhibit  $p$  through a complex priority model.

**OBIP  $\leftrightarrow$  WBSOS:** By the same argument as above, Example 14 shows that OBIP does not have weak full expressiveness w.r.t. WBSOS. The following example proves that they are incomparable.

*Example 16* Consider the same atomic component  $C$  and the same  $o_1$  as in Example 15. Component  $C$  and the behaviour of  $o_1\langle C \rangle$  are shown in Figure 9.

Consider the offer BIP glue operator  $o_3 = (\{p, r\}, \gamma_3)$ , with  $\gamma_3 = \{\dot{p}r\}$  applied to the composed component  $o_1\langle C \rangle$ . Both ports  $p$  and  $r$  are offered in the state 1, thus the interaction  $\dot{p}r$  is enabled in this state despite the fact that the state does not have any outgoing transition involving  $r$ . In the state 2,  $r$  is not offered and the interaction is not enabled.

As in Example 13, it is not possible to construct a hierarchy of witness BIP-like SOS operators corresponding to  $o_3$ , since the states 1 and 2 have the same sets of outgoing transitions in  $\sigma(o_1\langle C \rangle)$  and are, therefore, indistinguishable by witness BIP-like SOS operators.

#### 4.3 Additional results for flat systems

In the previous section, we have analysed the expressiveness relations among the introduced frameworks. In particular, we have shown that WBSOS, ABIP and OBIP are pairwise incomparable. It is interesting to observe that, when restricted to flat systems, i.e. those consisting of one operator applied to sets of atomic components, a comparison can be established: every OBIP operator can be expressed as an ABIP operator and every ABIP operator can be expressed as a WBSOS one.

The restriction to atomic components is crucial here: indeed, the equivalence  $q \uparrow p \Leftrightarrow \exists a : q \xrightarrow{a} \wedge p \in a$  holds only on atomic components.

**Proposition 8** *When restricted to flat systems, every OBIP operator can be expressed as an ABIP one:*

$$\forall n \in \mathbb{N}, \forall o \in \mathbf{OBIP}^n, \exists \tilde{o} \in \mathbf{ABIP}^n : \\ \forall C_1, \dots, C_n \in \mathbf{C}, \sigma(\tilde{o}\langle C_1, \dots, C_n \rangle) \simeq \sigma(o\langle C_1, \dots, C_n \rangle). \quad (35)$$

*Proof.* Consider an OBIP glue operator  $o = ((P_i)_{i=1}^n, \gamma)$ , with  $\gamma$  an extended interaction model on  $P = \bigcup_{i=1}^n P_i$ . For a given extended interaction  $a \in \gamma$  we denote  $I_a = \{b \in \gamma \mid \mathbf{fire}(b) = \mathbf{fire}(a)\}$  the set of all interactions in  $\gamma$  with the same firing support as  $a$ . Consider the following Boolean formula on  $P$ :

$$\phi_a = \bigwedge_{p \in \mathbf{fire}(a)} \mathbf{p} \wedge \bigvee_{b \in I_a} \left( \bigwedge_{p \in \mathbf{act}(b)} \mathbf{p} \wedge \bigwedge_{p \in \mathbf{neg}(b)} \bar{\mathbf{p}} \right).$$

For any set of atomic components  $C_1, \dots, C_n \in \mathbf{C}$  with  $\sigma(\gamma\langle C_1, \dots, C_n \rangle) = (Q, P, \rightarrow, \uparrow)$ , and for any state  $q \in Q$ , the valuation  $v_q = (\mathbf{p} = q \uparrow p)_{p \in P}$  satisfies  $\phi_a$  (denoted  $v_q \models \phi_a$ ) if and only if some extended interaction from  $I_a$  is enabled in the state  $q$ .

For a given valuation  $v \models \phi_a$ , we build an interaction with activation

$$b_a^v = \{\dot{p} \mid p \in \mathbf{fire}(a)\} \cup \{p \mid v(\mathbf{p}) = \text{true} \wedge p \notin \mathbf{fire}(a)\}$$

and an associated set of priorities

$$\pi_a^v = \{b_a^v < \{c\} \mid c \in 2^P \wedge \exists p \in c : v(p) = \text{false}\}.$$

The activation BIP glue operator equivalent to  $\gamma$  is  $\tilde{\sigma} = ((P_i)_{i=1}^n, \gamma', \pi')$ , with

- an interaction model with activation  $\gamma' = \{b_a^v \mid a \in \gamma, v \models \phi_a\}$ ,
- a complex priority model  $\pi' = \bigcup_{a \in \gamma} \bigcup_{v \in \text{Part}_a} \pi_a^v$ .

Consider a set of atomic components  $C_1, \dots, C_n \in \mathbf{C}$  and suppose that  $q \xrightarrow{\text{fire}(a)}$  in  $\sigma(\gamma\langle C_1, \dots, C_n \rangle)$  for some  $a \in \gamma$ . Let  $v_q = (\mathbf{p} = q \uparrow p)_{p \in P}$  be the valuation induced by the offer predicate in the state  $q$ . We have  $v_q \models \phi_a$ . By construction of  $\gamma'$ , we have  $b = \{p \mid p \in \text{fire}(a)\} \cup \{p \mid v_q(\mathbf{p}) = \text{true} \wedge p \notin \text{fire}(a)\} \in \gamma'$ . Notice that all ports of  $b$  are offered in the state  $q$  and  $\text{fire}(b) = \text{fire}(a)$ . Thus,  $q \xrightarrow{\text{fire}(b)}$  in  $\sigma(\gamma'\langle C_1, \dots, C_n \rangle)$ . By construction of  $\pi'$ , all interactions with higher priority contain a port  $p$ , such that  $v_q(\mathbf{p}) = \text{false}$ , i.e.  $p$  is not offered in the state  $q$  and, consequently, none of these interactions is enabled in  $q$ . Thus,  $q \xrightarrow{\text{fire}(a)}$  in  $\sigma(\pi'\gamma'\langle C_1, \dots, C_n \rangle)$ .

Suppose now that  $q \xrightarrow{\text{fire}(b)}$  in  $\sigma(\pi'\gamma'\langle C_1, \dots, C_n \rangle)$  for some  $b \in \gamma'$  and take  $v = (\mathbf{p} = p \in \text{fire}(b) \cup \text{act}(b))_{p \in P}$ . Since  $b$  is enabled, we have  $v(\mathbf{p}) \Rightarrow q \uparrow p$ . Suppose that, for some  $p \in P$ , we have  $q \uparrow p \wedge \neg v(\mathbf{p})$ . Since all the components are atomic, there is an interaction  $c$  enabled in  $q$  such that  $p \in c$ . By construction of  $\pi'$ , there is a priority rule  $b < \{c\}$  that would block  $b$  in the state  $q$ , contradicting the assumption  $q \xrightarrow{\text{fire}(b)}$ . Hence, we have  $v(\mathbf{p}) \Leftrightarrow q \uparrow p$ , for all  $p \in P$ . Let  $I_b = \{a \in \gamma \mid \text{fire}(a) = \text{fire}(b)\}$  and a Boolean formula  $\phi_b$  built as above. By construction,  $v \models \phi_b$ . Since  $v(\mathbf{p}) \Leftrightarrow q \uparrow p$ , this means that some interaction from  $I_b$  is enabled in the state  $q$  of  $\sigma(\gamma\langle C_1, \dots, C_n \rangle)$  and, therefore,  $q \xrightarrow{\text{fire}(b)}$  in  $\sigma(\gamma\langle C_1, \dots, C_n \rangle)$ .  $\square$

Despite the fact that ABIP does not have strong full expressiveness w.r.t. OBIP, for any system obtainable in OBIP from a given set of atomic components, we can build an equivalent one in ABIP from the same set of atomic components. Indeed, since FNASOS (hence, also OBIP) possesses uniform flattening, any hierarchy of OBIP operators can be replaced by a single operator. By Proposition 8, there exists a corresponding one in ABIP. These operators applied to the same set of atomic components would result in equivalent systems.

A WBSOS composition operator equivalent to a given ABIP operator  $((P_i)_{i=1}^n, \gamma, \pi)$  on atomic components can be constructed as follows. Given an interaction with activation  $a \in \gamma$ , we consider a set of witness BIP-like SOS rules  $S_a$  obtained by removing all redundant rules from the set

$$\left\{ \begin{array}{l} \left\{ q_i \xrightarrow{\text{fire}(a) \cap P_i} q'_i \mid i \in I \right\} \\ \left\{ q_i = q'_i \mid i \notin I \right\} \\ \left\{ q_{j(\alpha)} \xrightarrow{b(\alpha) \cap P_{j(\alpha)}} \mid \alpha \in \pi(a) \right\} \\ \left\{ q_k \xrightarrow{c \cap P_k} \mid c \in J, k \in K_c \right\} \\ \hline q_1 \dots q_n \xrightarrow{\text{fire}(a)} q'_1 \dots q'_n \end{array} \middle| \begin{array}{l} J \subseteq 2^P, \text{ s.t. } \text{act}(a) \subseteq \bigcup_{c \in J} c; \\ b : \pi(a) \Rightarrow \bigcup \pi(a) : (\alpha).b \in \alpha, \\ j : \pi(a) \Rightarrow [1, n] : (\alpha).b(\alpha) \cap P_j \neq \emptyset \end{array} \right\}, \quad (36)$$

where  $I = \{i \in [1, n] \mid \mathbf{fire}(a) \cap P_i \neq \emptyset\}$  and  $K_c = \{k \in [1, n] \mid c \cap P_k \neq \emptyset\}$ . In the positive premises we require a set of transitions  $J$  that has to be enabled in the state of the composed component such that each port from the activation support of  $a$  is involved in at least one transition and, consequently, offered (since we only consider atomic components here). In the negative premises, we require all higher-priority sets of interactions to be disabled.

**Proposition 9** *Let  $((P_i)_{i=1}^n, \gamma, \pi)$  be an ABIP operator and take a WBSOS operator  $o = ((P_i)_{i=1}^n, \bigcup_{a \in \gamma} S_a)$ , with  $S_a$  the set of rules (36), for each  $a \in \gamma$ . Then, for any set of atomic components  $C_1, \dots, C_n$ , we have  $\sigma(\pi\gamma\langle C_1, \dots, C_n \rangle) = \sigma(o\langle C_1, \dots, C_n \rangle)$ .*

*Proof.* Let  $q \xrightarrow{d}$  in  $\sigma(\pi\gamma\langle C_1, \dots, C_n \rangle)$ . There exists an interaction with activation  $a \in \gamma$  such that  $\mathbf{fire}(a) = d$ ,  $q \uparrow p$  holds for all  $p \in \mathbf{act}(a)$ , and none of the higher-priority sets of interactions  $\alpha \succ a$  are enabled. Consider the set of interactions  $J$  that are enabled in the state  $q$  of the unrestricted parallel composition of  $C_1, \dots, C_n$ , i.e.  $J = \{c \subseteq P \mid (c \cap P_i = \emptyset) \vee (q_i \xrightarrow{c \cap P_i} \text{ in } \sigma(C_i)), i \in [1, n]\}$ . Notice that  $\mathbf{act}(a) \subseteq \bigcup_{c \in J} c$ , since all ports in  $\mathbf{act}(a)$  are offered. By construction of the operator  $o$ , there is a subset of rules in  $S_a$  with the activation premises defined by  $J$ . Since all interactions in  $J$  are enabled, these activations premises are satisfied in  $q$ . Since none of the sets  $\alpha \succ a$  is enabled, we can choose, for each  $\alpha$ , the corresponding  $b_\alpha \in \alpha$  and  $j_\alpha \in [1, n]$ , such that  $q_{j_\alpha} \xrightarrow{b_\alpha \cap P_{j_\alpha}}$ . By taking, in (36), the choice function realisations  $b : \alpha \mapsto b_\alpha$  and  $j : \alpha \mapsto j_\alpha$ , we obtain a rule in  $S_a$ <sup>8</sup>, whereof the negative premises are also satisfied and we have  $q \xrightarrow{d}$  in  $\sigma(o\langle C_1, \dots, C_n \rangle)$ .

Let  $q \xrightarrow{d}$  in  $\sigma(o\langle C_1, \dots, C_n \rangle)$ . There is a rule  $r \in S_a$ , for some  $a \in \gamma$ , that allows this transition in the composed component behaviour. Let  $J$  be the set of interactions defining the activation premises in  $r$ . Notice that all interactions  $c \in J$  are enabled in the state  $q$ . By construction of  $r$ ,  $\mathbf{fire}(a) = d$ . For all ports  $p \in \mathbf{act}(a)$ , there is an interaction  $c \in J$ , such that  $p \in c$ . Hence,  $q \uparrow p$ . For each set of interactions  $\alpha \succ a$ , there is one negative premise in  $r$ . Since this premise is satisfied,  $\alpha$  is disabled. Thus, the interaction  $a$  is enabled in the state  $q$  and  $q \xrightarrow{d}$  in  $\sigma(\pi\gamma\langle C_1, \dots, C_n \rangle)$ .  $\square$

## 5 Related Work

The results in this paper build mainly on our own previous work. However, the following related work should also be mentioned.

Usually, formalisms are compared by flattening structure and reducing to a behaviourally equivalent model, e.g. automata and Turing machines. In this manner, all finite state formalisms turn out to be expressively equivalent independently of the features used for the composition of behaviours. Many models and languages are Turing-expressive, while their coordination capabilities are tremendously different. [11]

<sup>8</sup> The rule  $S_a$  might be removed from the operator  $o$  as redundant, however this would imply the existence of another rule with the same conclusion and whereof premises would also be satisfied in the state  $q$ .

A first framework formally capturing meanings of expressiveness for sequential programming languages and taking into account not only the semantics but also the primitives of languages was provided in [21]. It allows formal reasoning about and distinguishing between *core elements* of a language and *syntactic sugar*. Although a number of studies have taken a similar approach in the context of concurrency, we will point only to [23] and the references therein. The key difference of our approach lies in the strong separation between the computation and coordination aspects of the behaviour of concurrent systems. Indeed, we consider that all sequential computation resides within the components of the system that are not subject to any kind of modification. Thus, we focus on the following question: *what system behaviour can be obtained by coordination of a given set of concurrent components?*

An extensive overview of SOS formats is provided in [31], including some results comparing their expressiveness. More results comparing different formats of SOS can be found in [30]. The expressiveness property is closely related to the translation between languages. One of the definitions of encoding compared with other approaches can be found in [22]. It should be noted, however, that the above mentioned separation of concerns principle also leads to a very simple rule format. Indeed, the format that we consider is essentially a small subset of GSOS. Our focus in this paper, is more on the expressiveness of the coordination mechanism provided by BIP than on that of the various SOS rule features.

There exist several works comparing BIP with various connector frameworks. A comparative study of three connector frameworks—tile model [16], wire calculus [36] and BIP [8]—was presented in [17]. An attempt to relate BIP and Reo has been made recently [19]. From the operational semantics perspective, these comparisons only take in account operators with positive premises. In particular, priority in BIP is not considered, whereas it is central for the present paper.

Finally, in our formalisation of component-based frameworks, we rely on the notion of “behaviour type”. This can cover a very large spectrum, ranging from programs and labelled transition systems, through OSGi bundles and browser plug-ins, to systems of differential equations etc. Behaviour types can be organised in type systems and studied separately, as, for example, in the co-algebra theory [34]. However, this notion should be distinguished, for instance, from classes in object-oriented programming or session [18, 26] and behavioural [27] types for communication protocols. For instance, the notion of a class could be compared to that of a behaviour type in our sense as follows: a program would typically comprise a multitude of classes, whereas a component framework has only one underlying behaviour type. Although, in principle, component-based frameworks can be heterogeneous, e.g. Ptolemy II [20], that is rely on several distinct behaviour types for the design process, those aimed at the design of executable systems must have an underlying unifying behaviour type allowing the study and manipulation of a system as a whole.

## 6 Conclusion

Our previous investigations [4] of several properties that we consider fundamental for component-based design frameworks have revealed that the often encountered informal statement: “BIP possesses the expressiveness of the universal glue” (or its



equivalent in slightly different formulations) is based on an erroneous proposition in previous work [11, Proposition 4]. We have, therefore, undertaken an additional study of BIP expressiveness, whereof the results have been presented in this paper.

To achieve this goal, we rely on the algebraic formalisation of the notion of component-based design framework introduced in [4] and refined in this paper. We have defined two new comparison relations among component-based frameworks, *weak* and *strong full expressiveness*, which characterise whether systems that can be constructed in one framework can also be constructed in the other. Although these properties are stronger than the corresponding expressiveness preorders introduced in [11], they provide a simpler setup for the comparison of the expressiveness of component-based frameworks. The notion of weak full expressiveness is different from the *weak expressiveness preorder* in [11]: the former relaxes the strong form of the property by allowing hierarchical composition of glue operators, whereas the latter considers only flat operators, but allows a limited use of additional coordinating behaviour. Studying the combination of the two relaxations could be an interesting direction for future work.

We have studied the weak and full expressiveness of BIP w.r.t. operators defined by SOS rules in a particular format, which we call *BIP-like SOS*, and some of its restrictions. The set of all the operators that can be defined using BIP-like SOS rules is the “universal glue”, w.r.t. which full expressiveness has been erroneously claimed in [11]. The two restrictions that we consider are characterised by the so-called *inhibiting relation*, which identifies the interactions, enabledness whereof inhibits that of another given one. The first restriction, SiBSOS, considers only sets of BIP-like SOS rules with *simple* inhibiting relations, where the enabledness of one interaction is always sufficient to inhibit another one. To the contrary, operators defined by sets of rules with *complex* inhibiting relations might require a combination of interactions to be enabled simultaneously to inhibit one given interaction.

Semantics of BIP operators is defined in terms of rules with only simple inhibiting relations. We observe, however, that there are two obstacles to achieving strong full expressiveness w.r.t. SiBSOS: 1) a BIP priority model is required to be a *strict partial order* on interactions and 2) by the definition of the BIP operational semantics, priorities can only be applied to interactions that appear in the interaction model. The combination of these two requirements ensures that priorities cannot introduce new deadlocks. However, negative premises in BIP-like SOS rules—which correspond to priorities in BIP glue operators—can do so. In order to introduce deadlocks, the inhibiting relation must have cycles. We show that BIP glue operators have weak full expressiveness w.r.t. BIP-like SOS operators that have acyclic inhibiting relations, with at most  $d + 1$  layers of glue necessary to encode a BIP-like SOS operator, whereof the depth of the inhibiting relation is  $d$ .

A relaxation of both of the above requirements together recovers *strong* full expressiveness w.r.t. SiBSOS. However, it calls for a definition of the operational semantics of BIP glue operators, which combines the interaction and the priority models, as opposed to the classical definition, where the interaction model is applied first, then the priority model is applied to the resulting component.

A relaxation of only the first requirement, which does not require any other modifications of the BIP semantics, leads to *weak* full expressiveness w.r.t. SiBSOS.

Moreover, we have shown that at most two layers of glue are necessary to encode any operator.

In order to achieve the expressive power of the full BIP-like SOS format, we have defined an extension of BIP, called XBIP, which allows priorities to use combinations of interactions.

In order to put the above results into a broader perspective, we have also analysed the expressiveness relations among several variations of BIP semantics and SOS formats used to define these. In particular, we have considered two new types of premises that can be used to define glue operators. The offer predicate  $q \uparrow p$  (and its negative form  $q \nmid p$ ) is used to access information about the activation of the port  $p$  in the current state of the corresponding atomic component. Another variation, witness BIP-like SOS, using premises of the form  $q \xrightarrow{a}$ , allows testing the enabledness of an interaction in the current global state without firing it.

The offer predicate is used to define the homonymous, offer semantics of BIP, introduced in [13] and studied extensively in [4]. The Offer BIP turns out to be largely incomparable to other variations of BIP considered in the paper. The advantages of offer BIP, as outlined in [4], are that it possesses a number of nice properties, in particular strong full expressiveness w.r.t. the rule format used to define its semantics and flattening. More importantly, contrary to classical BIP and the variations obtained by adding positive premises, offer BIP lends itself nicely to a Boolean encoding and, therefore, efficient implementations using, for instance, Binary Decision Diagrams as in JavaBIP [9].

As mentioned above, studying the combination of the two weak forms of full expressiveness—allowing both hierarchical glue and limited use of additional co-ordinating behaviour—could be an interesting direction for future work. Another direction for future work would consist in exploring the expressiveness of the full BIP framework, including the data manipulation and transfer as formalised in [14].

## Acknowledgments

The contribution of Simon Bliudze to the work presented in this paper has been partially funded by Hauts-de-France regional STaRS programme under the grant agreement № 2019.00170.

## References

1. Boulanger et al., J.L.: *SCADE: Language and Applications*, 1st edn. Wiley-IEEE Press (2015)
2. Attie, P., Baranov, E., Bliudze, S., Jaber, M., Sifakis, J.: A general framework for architecture composability. In: D. Giannakopoulou, G. Salaün (eds.) 12th International Conference on Software Engineering and Formal Methods (SEFM 2014), no. 8702 in LNCS, pp. 128–143. Springer International Publishing, Switzerland (2014). DOI 10.1007/978-3-319-10431-7\_10
3. Attie, P., Baranov, E., Bliudze, S., Jaber, M., Sifakis, J.: A general framework for architecture composability. *Formal Aspects of Computing* **18**(2), 207–231 (2016). DOI 10.1007/s00165-015-0349-8. Open access
4. Baranov, E., Bliudze, S.: Offer semantics: Achieving compositionality, flattening and full expressiveness for the glue operators in BIP. *Science of Computer Programming* **109**(0), 2–35 (2015). DOI 10.1016/j.scico.2015.05.011. Selected Papers of the 6th Interaction and Concurrency Experience (ICE 2013)

5. Baranov, E., Bliudze, S.: A note on the expressiveness of BIP. In: Proceedings of the Combined 23rd International Workshop on Expressiveness in Concurrency and 13th Workshop on Structural Operational Semantics, EXPRESS/SOS 2016, *EPTCS*, vol. 222, pp. 1–14 (2016). DOI 10.4204/EPTCS.222.1
6. Basten, T.: Branching bisimilarity is an equivalence indeed! *Information Processing Letters* **58**(3), 141–147 (1996)
7. Basu, A., Bensalem, S., Bozga, M., Combaz, J., Jaber, M., Nguyen, T.H., Sifakis, J., et al.: Rigorous component-based system design using the BIP framework. *IEEE Software* **28**(3), 41–48 (2011). DOI 10.1109/MS.2011.27
8. Basu, A., Bozga, M., Sifakis, J.: Modeling heterogeneous real-time components in BIP. In: 4<sup>th</sup> IEEE Int. Conf. on Software Engineering and Formal Methods (SEFM06), pp. 3–12 (2006). DOI 10.1109/SEFM.2006.27. Invited talk
9. Bliudze, S., Mavridou, A., Szymanek, R., Zolotukhina, A.: Exogenous coordination of concurrent software components with JavaBIP. *Software: Practice and Experience* **47**(11), 1801–1836 (2017). DOI 10.1002/spe.2495
10. Bliudze, S., Sifakis, J.: The algebra of connectors — Structuring interaction in BIP. In: Proc. of the EMSOFT’07, pp. 11–20. ACM SigBED (2007). DOI 10.1145/1289927.1289935
11. Bliudze, S., Sifakis, J.: A notion of glue expressiveness for component-based systems. In: F. van Breugel, M. Chechik (eds.) CONCUR 2008, *LNCS*, vol. 5201, pp. 508–522. Springer (2008). DOI 10.1007/978-3-540-85361-9\_39
12. Bliudze, S., Sifakis, J.: Causal semantics for the algebra of connectors. *Formal Methods in System Design* **36**(2), 167–194 (2010). DOI 10.1007/s10703-010-0091-z
13. Bliudze, S., Sifakis, J.: Synthesizing glue operators from glue constraints for the construction of component-based systems. In: S. Apel, E. Jackson (eds.) 10<sup>th</sup> International Conference on Software Composition, *LNCS*, vol. 6708, pp. 51–67. Springer (2011). DOI 10.1007/978-3-642-22045-6\_4
14. Bliudze, S., Sifakis, J., Bozga, M.D., Jaber, M.: Architecture internalisation in BIP. In: Proceedings of the 17th International ACM Sigsoft Symposium on Component-based Software Engineering, CBSE ’14, pp. 169–178. ACM, New York, NY, USA (2014). DOI 10.1145/2602458.2602477
15. Bloom, B.: Ready simulation, bisimulation, and the semantics of CCS-like languages. Ph.D. thesis, Massachusetts Institute of Technology (1989)
16. Bruni, R., Lanese, I., Montanari, U.: A basic algebra of stateless connectors. *Theor. Comput. Sci.* **366**(1), 98–120 (2006). DOI 10.1016/j.tcs.2006.07.005
17. Bruni, R., Melgratti, H., Montanari, U.: Connector algebras, petri nets, and bip. In: E. Clarke, I. Virbitskaite, A. Voronkov (eds.) Perspectives of Systems Informatics, *Lecture Notes in Computer Science*, vol. 7162, pp. 19–38. Springer Berlin Heidelberg (2012). DOI 10.1007/978-3-642-29709-0\_2
18. Dezani-Ciancaglini, M., de’Liguoro, U.: Sessions and session types: An overview. In: C. Laneve, J. Su (eds.) Web Services and Formal Methods: 6th International Workshop, WS-FM 2009, no. 6194 in *Lecture Notes in Computer Science*, pp. 1–28. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-14458-5\_1
19. Dokter, K., Jongmans, S.T.Q., Arbab, F., Bliudze, S.: Relating BIP and reo. In: S. Knight, I. Lanese, A. Lluch-Lafuente, H.T. Vieira (eds.) Proceedings 8th Interaction and Concurrency Experience, ICE 2015, Grenoble, France, 4-5th June 2015., *EPTCS*, vol. 189, pp. 3–20 (2015). DOI 10.4204/EPTCS.189.3
20. Eker, J., Janneck, J., Lee, E., Liu, J., Liu, X., Ludvig, J., Neuendorffer, S., Sachs, S., Xiong, Y.: Taming heterogeneity: The Ptolemy approach. *Proceedings of the IEEE* **91**(1), 127–144 (2003). DOI 10.1109/JPROC.2002.805829
21. Felleisen, M.: On the expressive power of programming languages. In: 3<sup>rd</sup> European Symposium on Programming (ESOP’90), *LNCS*, vol. 432, pp. 134–151. Springer (1990). DOI 10.1007/3-540-52592-0\_60
22. van Glabbeek, R.J.: Musings on encodings and expressiveness. In: B. Luttik, M.A. Reniers (eds.) Proceedings Combined 19th International Workshop on Expressiveness in Concurrency and 9th Workshop on Structured Operational Semantics, EXPRESS/SOS 2012, Newcastle upon Tyne, UK, September 3, 2012., *EPTCS*, vol. 89, pp. 81–98 (2012). DOI 10.4204/EPTCS.89.7
23. Gorla, D.: Towards a unified approach to encodability and separation results for process calculi. *Information and Computation* **208**(9), 1031–1053 (2010). DOI 10.1016/j.ic.2010.05.002

24. Gößler, G., Sifakis, J.: Priority systems. In: F.S. de Boer, M.M. Bonsangue, S. Graf, W.P. de Roever (eds.) *Formal Methods for Components and Objects*, Second International Symposium, FMCO 2003, Leiden, The Netherlands, November 4-7, 2003, Revised Lectures, *Lecture Notes in Computer Science*, vol. 3188, pp. 314–329. Springer (2003). DOI 10.1007/978-3-540-30101-1\_15
25. Gössler, G., Sifakis, J.: Composition for component-based modeling. *Science of Computer Programming* **55**(1–3), 161–183 (2005). DOI 10.1016/j.scico.2004.05.014. *Formal Methods for Components and Objects: Pragmatic aspects and applications*
26. Honda, K., Vasconcelos, V.T., Kubo, M.: Language primitives and type discipline for structured communication-based programming. In: C. Hankin (ed.) *Programming Languages and Systems, LNCS*, vol. 1381, pp. 122–138. Springer Berlin Heidelberg (1998). DOI 10.1007/BFb0053567
27. Hüttel, H., Lanese, I., Vasconcelos, V.T., Caires, L., Carbone, M., Deniélou, P.M., Mostrous, D., Padovani, L., Ravara, A., Tuosto, E., Vieira, H.T., Zavattaro, G.: Foundations of session types and behavioural contracts. *ACM Comput. Surv.* **49**(1), 3:1–3:36 (2016). DOI 10.1145/2873052
28. Mavridou, A., Stachtiri, E., Bliudze, S., Ivanov, A., Katsaros, P., Sifakis, J.: Architecture-based design: A satellite on-board software case study. In: 13th International Conference on Formal Aspects of Component Software (FACS 2016), *Lecture Notes in Computer Science*, vol. 10231, pp. 260–279 (2016). DOI 10.1007/978-3-319-57666-4\_16
29. Milner, R.: *Communication and Concurrency*. Prentice Hall International Series in Computer Science. Prentice Hall (1989)
30. Mousavi, M., Phillips, I., Reniers, M.A., Ulidowski, I.: Semantics and expressiveness of ordered SOS. *Information and Computation* **207**, 85–119 (2009). DOI 10.1016/j.ic.2007.11.008
31. Mousavi, M., Reniers, M.A., Groote, J.F.: SOS formats and meta-theory: 20 years after. *Theoretical Computer Science* **373**(3), 238–272 (2007). DOI 10.1016/j.tcs.2006.12.019
32. Park, D.M.R.: Concurrency and automata on infinite sequences. *Proceedings of the 5th GI-Conference on Theoretical Computer Science* pp. 167–183 (1981). DOI 10.1007/BFb0017309
33. Plotkin, G.D.: A structural approach to operational semantics. Tech. Rep. DAIMI FN-19, University of Aarhus (1981). URL <http://citeseer.ist.psu.edu/plotkin81structural.html>
34. Rutten, J.J.M.M.: Universal coalgebra: a theory of systems. *Theor. Comput. Sci.* **249**(1), 3–80 (2000). DOI 10.1016/S0304-3975(00)00056-6
35. Sifakis, J.: A framework for component-based construction. In: 3<sup>rd</sup> IEEE Int. Conf. on Software Engineering and Formal Methods (SEFM05), pp. 293–300 (2005). DOI 10.1109/SEFM.2005.3. Keynote talk
36. Sobocinski, P.: A non-interleaving process calculus for multi-party synchronisation. In: F. Bonchi, D. Grohmann, P. Spoletini, E. Tuosto (eds.) *ICE, EPTCS*, vol. 12, pp. 87–98 (2009). DOI 10.4204/EPTCS.12.6
37. Stachtiri, E., Mavridou, A., Katsaros, P., Bliudze, S., Sifakis, J.: Early validation of system requirements and design through correctness-by-construction. *Journal of Systems and Software* **145**, 52–78 (2018). DOI <https://doi.org/10.1016/j.jss.2018.07.053>